



A numerical study of F1 cornering aerodynamics using OpenFOAM

Master thesis
Center for Hydro and Aero laboratory-HEPIA

Author: Amirmohammad Rajabi
Professor: Prof. Flavio Noca
Place and date: Lausanne, September 2023

Abstract

The introduction of new regulations for the 2022 season of Formula 1 brought about considerable changes to how the aerodynamics of the cars work to generate downforce. In addition to front and rear wings, the cars now rely on the ground-effect aerodynamics to generate downforce. As a result of these changes, the shape of the floor and its interaction with the ground becomes an important factor in the car performance. Given the vast changes in the aerodynamics of the new cars, the teams will have to re-evaluate and analyse the behaviour of the cars in different parts of the track. Understanding the behaviour of the car during cornering is of paramount importance since it is in these parts that aerodynamic performance is the most critical [1]. Various experimental attempts have been made to accurately study cornering aerodynamics. However, these methods come with their own limitations. To this end Center for Hydro and Aero laboratory of HEPIA has developed a small scale wind tunnel with a rotating belt to study this phenomena. Given the flexibility of numerical methods, it was deemed beneficial to build a numerical platform for the experimental setup. In this work the latter was done using OpenFOAM. The Navier-Stokes equations were solved using the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm. The turbulent flow was characterized by the $k - \omega$ *SST* model. The results of the simulation demonstrated that the model behaves according to standard aerodynamics of F1 cars. However, it was observed that due to unsteady nature of the flow and the limitation of the computational domain certain aerodynamic parameters such as lift show oscillatory behaviour. Additionally, certain behaviours inherent to the model's design were observed. One such behaviour was the -0.4 pressure coefficient at the junction of the halo and the body due to the housing created for measurement devices and the arm. Furthermore, the PIV data and simulation data were compared. It was shown that the simulation data has a good agreement with the flow observed in the tunnel. However, the difference in maximum velocity between PIV and simulation in one case was up to 25%. The latter was deemed to be due to experimental errors since on the second set of data with improved accuracy the error was reduced to 11%.

Keywords: Aerodynamics, Cornering, Formula 1, Wind tunnel, Curved flow, PIV

Acknowledgement

I am grateful to Prof. Noca for his guidance and providing me with the opportunity to conduct this project. Additionally, I would like to thank Prof. Gallaire for agreeing to be EPFL's supervisor for this project.

I would like to also thank Phd student Baptiste Gousset for his valuable help with technical and administrative aspects of the project. Additionally, a special thanks goes to Malicia Leipold and Dmytro Yakymenko for conducting PIV experiments for this project.

Contents

Abstract	i
Acknowledgement	iii
Contents	iv
List of Figures	ix
List of Tables	xi
Nomenclature	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Goal and objectives	1
1.3 Literature review	2
1.3.1 Techniques for Aerodynamic Analysis of Cornering Vehicles [1] . .	2
1.3.1.1 Experimental methods	2
1.3.1.2 Numerical methods	3
2 Methodology	5
2.1 Geometry	5
2.1.1 Car model and computational domain	5
2.1.2 Model simplifications	7
2.1.2.1 Suspensions	8
2.1.2.2 Wheels	8
2.1.3 Front and rear wing	9
2.1.4 Element names	10
2.2 Governing equations	13
2.2.1 Continuity equation	13
2.2.2 Momentum equation	14
2.2.3 Turbulence model	14
2.3 Computational setup	15
2.3.1 Solver choice	15

Contents

2.3.2	Boundary conditions	16
2.3.2.1	U and p boundary conditions	16
2.3.2.2	ω , k , and nut conditions	18
2.3.3	Solver schemes	19
2.3.3.1	Gradient schemes	19
2.3.3.2	Divergence schemes	19
2.3.3.3	Surface normal gradient and Laplacian schemes	20
2.3.3.4	Interpolation	21
2.3.4	Solution schemes	21
2.3.4.1	Solvers	21
2.3.4.2	SIMPLE settings	22
2.3.4.3	Under-relaxation factors	22
2.4	Mesh	23
2.4.1	Mesh convergence study	23
2.4.2	Boundary layer (BL)	27
2.4.3	Grid independence criteria	29
2.4.4	Mesh quality	29
2.4.5	Results	31
2.4.5.1	Lift and drag coefficients	31
2.4.5.2	Y+ value	32
2.4.6	Choice of final mesh	32
3	Results and discussion	35
3.1	Convergence	35
3.1.1	Residuals	35
3.1.2	Lift and drag coefficients	35
3.2	Inlet and ground	39
3.3	velocity contours	40
3.3.1	Front wing	40
3.3.2	Diffuser	40
3.3.3	Rear wing	42
3.3.4	Floor	43
3.4	Pressure coefficient	44
3.5	PIV comparison	46
3.5.1	Plane 1: wake of the car	47
3.5.2	Plane 2: wake of the car	48
4	Conclusion	51
4.1	Summary	51
4.2	Future work	52

A	Appendix	57
A.1	$k - \omega$ SST variables [2] [3]	57
A.2	decomposeParDict settings	59
A.3	ControlDict settings	59
A.4	Boundary conditions	62
A.4.1	U	62
A.4.2	p	65
A.4.3	k	68
A.4.4	omega	71
A.4.5	nut OpenFOAM v2112	75
A.4.6	nut OpenFOAM v2212	78
A.5	fvSchemes	82
A.6	fvSolutions	83
A.7	Y+ value	85

List of Figures

1.1	Curved computational domain (Top) and non-inertial domain (Bottom)	3
2.1	Comparison of the initial (Top) and final (Bottom) car models.	6
2.2	Position of the car within the tunnel. The dimensions take into account the adjustments made to the wheels.	7
2.3	3D representation of the computational domain.	7
2.4	Comparison of the initial (Left) and final (Right) suspension heads.	8
2.5	Image of the removed suspension.	8
2.6	Image of the old (Left) and new (Right) rear suspension.	9
2.7	Comparison of the initial (Left) and final (Right) state of the wheels.	9
2.8	Fillet sizing of the rear wing elements.	10
2.9	Front wing joints indicated by red circles.	10
2.10	Name selection of various car elements.	11
2.11	Wing spaces are four small fillets at the intersection of the beam wings and the exhaust highlighted in red.	11
2.12	Elements named chassis spaces are small surfaces along the edge of the chassis and floor (Top) and two fillets on both sides of the exhaust (Bottom).	12
2.13	Winglets group is composed of the two larger winglets behind the front wing (Top) and two smaller winglets connected to the endplates (Bottom).	12
2.14	The name selection group called radii is created under each wheel.	13
2.15	Name selection of the computational domain (viewed from above). The top surface of the tunnel is also part of the tunnel walls group.	13
2.16	SIMPLE algorithm flow chart.	16
2.17	Second order upwind differencing scheme setup in the case of mass flow-in (A) and flow-out (B).[4]	20
2.18	Bodies of influence placement within the tunnel.	24
2.19	User input interface for body of influence sizing (Top), face sizing (middle), and curvature sizing (Bottom).	24
2.20	Sizing of other mesh parts such as the ground and tunnel walls.	26
2.21	Rear wing surface mesh on the coarsest (Left) and finest (Right) mesh.	27
2.22	Front wing surface mesh on the coarsest (Left) and finest (Right) mesh.	27
2.23	Chassis surface mesh on the coarsest (Left) and finest (Right) mesh.	27
2.24	Front (Left) and rear (Right) wing boundary layer mesh.	28

List of Figures

2.25	Schematic of how mesh orthogonality is defined.[5]	29
2.26	Schematic of how mesh skewness is defined.[5]	30
2.27	Schematic of how mesh aspect ratio is defined.[5]	30
2.28	Average value of drag coefficient as a function of mesh elements.	31
2.29	Average value of lift coefficient as a function of mesh elements.	32
3.1	Residual curves of the final simulation. The sudden drops are a result of URF change.	36
3.2	Drag coefficient curves for each mesh.	37
3.3	Lift coefficient curves for each mesh.	38
3.4	Velocity contour of the inlet.	39
3.5	Velocity contour of the ground.	40
3.6	Velocity field behind the front wing.	41
3.7	Velocity field behind the rear bodywork.	41
3.8	Extruded housings on the floor which cause flow blockage.	42
3.9	Velocity contour of the rear wing wake.	43
3.10	Velocity stream lines near the ground leading to the wake of the rear wheel.	43
3.11	Rear wing velocity contour with streamline representation of the flow from the back (Left) and front (Right) of the car forming the outer part of the wake.	44
3.12	Rear wing velocity contour with streamline representation of the flow from the back of the car forming the inside of the wake.	44
3.13	Velocity field under the car.	45
3.14	Streamlines around the wheels and suspensions.	45
3.15	Pressure coefficient over the surface of the car.	46
3.16	Visualisation of streamlines around the halo and the cylindrical housing.	46
3.17	Pressure coefficient over the surface of the underbody.	47
3.18	Picture of the car according to plane 1 configurations.	47
3.19	Picture of the car according to plane 2 configurations.	48
3.20	Result of the PIV experiment (a) and CFD simulations (b) at plane 1.	49
3.21	Result of the PIV experiment (a) and CFD simulations (b) at plane 2.	50

List of Tables

2.1	Solver settings for simulation parameters.	22
2.2	Mesh sizing of the car and bodies of influence for mesh 1 and mesh 2. . . .	25
2.3	Mesh sizing of the car and bodies of influence for mesh 3 and mesh 4. . . .	25
2.4	Mesh sizing of the car and bodies of influence for mesh 5.	26
2.5	Boundary layer configuration for the car and computational domain. . . .	28
2.6	Mesh quality of the studied meshes.	30
2.7	Average value of lift and drag coefficients.	31
2.8	Y+ values of mesh 1.	33
A.1	Y+ values for mesh 2.	85
A.2	Y+ values for mesh 3.	85
A.3	Y+ values for mesh 4.	85
A.4	Y+ values for mesh 5.	85

Nomenclature

Acronyms

BL	Boundary Layer
BOI	Body of Influence
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
F1	Formula 1
FIA	Fédération Internationale de l'Automobile
PIV	Particle Image Velocimetry
RANS	Reynolds-Averaged Navier-Stokes
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
URF	Under-relaxation factors

Greek Symbols

κ	von Karman constant Non-dimensional
μ	Dynamic viscosity $\text{kgm}^{-1}\text{s}^{-1}$
ν	Kinematic viscosity m^2s^{-1}
ω	Angular velocity rads^{-1}
ρ	Density kgm^{-3}

Roman Symbols

$\boldsymbol{U}, \boldsymbol{v}$	Velocity vector ms^{-1}
A	Area m^2
C_d	Drag coefficient

Nomenclature

C_l	Lift coefficient
p, P	Pressure atm
r	Radius m
t	time s
V, v	velocity ms^{-1}
k	Turbulent kinematic energy m^2s^{-2}

Superscripts

b	body
d	drag
l	lift
log	log layer
t	turbulent
vis	viscous layer

1 Introduction

The introduction of new regulations for the 2022 season of Formula 1 brought about considerable changes to how the aerodynamics of the cars work to generate downforce. In addition to front and rear wings, the cars now rely on the ground-effect aerodynamics to generate downforce. As a result of these changes, the shape of the floor and its interaction with the ground becomes an important factor in the car performance. Given the vast changes in the aerodynamics of the new cars, the teams will have to re-evaluate and analyse the behaviour of the cars in different parts of the track. Understanding the behaviour of the car during cornering is of paramount importance since it is in these parts that aerodynamic performance is the most critical [1].

1.1 Motivation

At Center for Hydro and Aero laboratory of HEPIA, a test-bench has been built to analyse cornering aerodynamics of F1 cars. Given the flexibility of numerical methods, they can be used to experiment with new ideas before implementation. Therefore establishing a valid CFD model can be useful for making improvements to the current experimental setup and providing additional insight in areas where experiment might not be sufficient.

1.2 Goal and objectives

The goal of this project is to build a CFD platform for the current test bench to numerically analyse the flow around the car. Furthermore, the results of the simulation will be compared to the experimental PIV results.

1.3 Literature review

1.3.1 Techniques for Aerodynamic Analysis of Cornering Vehicles [1]

In this paper Keogh et Al. provide an overview on various experimental and numerical methods used to study cornering motion of vehicles.

Despite the importance of aerodynamics in cornering motion, in the automotive and motorsports industries vehicles are studied in a straight line configuration. For studying cornering motion, numerical methods are used. However the main shortcoming of the latter is their insensitivity to changing conditions.

1.3.1.1 Experimental methods

The most common method of studying cornering aerodynamics in wind tunnels is changing the yaw angle of the car. The latter is rather a simple yet cost efficient method. In order to conduct more realistic experiments numerous attempts have been made to build rigs adapted for cornering aerodynamics.

One of the first methods to study cornering motion was the whirling arm built by an English mathematician Benjamin Robins in 1746. The device recreates cornering motion by moving the model around a central point using an arm. One of the most important characteristics of this model is that the rotating object is the model itself which means for vehicle testing a moving ground will not be needed. However, the major shortcoming of the whirling arm is that the rotation of the arm leads to the car moving in its own wake resulting in high turbulence and swirling motion. Further attempts were made by UK's National Physics Laboratory (NPL) to reduce the swirling however they were not successful and produced limited experimental results.

The rotary rig is another possible experimental device to study cornering aerodynamics. In principle, such rig is used to study dynamic stability of aircrafts in wind tunnels. However, one type of motion on rotary rig called coning can recreate the same conditions as cornering. In this case the model is rotated around an axis parallel to the freestream flow. This motion helps avoid the car moving in its own wake. However, since the distance between the model and the walls is constantly changing large scale facilities have to be built to overcome flow interference with the walls rendering this method costly.

Curved test sections are another method of testing. However, the main problem of this method is the flow quality since the flow is forced to follow the shape of the walls. Additionally, since the outer walls are concave they create a pressure side resulting in a decelerated flow while the opposite happens on the inner walls which is the inverse

of cornering motion in freestream conditions. Additionally, this effect becomes more pronounced as the radius of the curvature is decreased.

Bent model is another experimental method. In this method a curved model is placed in freestream flow. The main problem of this model is the difficulty and feasibility of making the model itself. Additionally, for each corner a new model has to be built rendering this method inefficient.

1.3.1.2 Numerical methods

The most common method of simulating cornering conditions is the use of a non-inertial reference frame as the computational domain. In this case the domain is rotated about an external point. The distance between the center of rotation and the domain is determined according to the corner radius. Another way of characterizing the computational domain is to use a curved boundary. However, the main drawback of this method is the lack of flexibility in studying various corners since the mesh has to be re-adjusted for each corner. Figure 1.1 presents a schematic of the methods used in numerical studies.

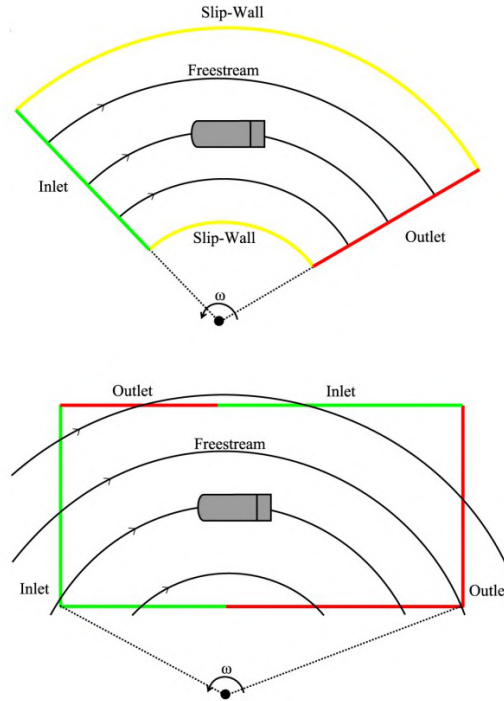


Figure 1.1: Curved computational domain (Top) and non-inertial domain (Bottom)

An important part of numerical simulation is the aerodynamic force analysis. When

simulating a curved flow all the forces and moments except drag remain the same. On a straight line the drag force acts in the opposite direction of the movement however in a curved flow this force acts on the car along the curvature which means the angle is constantly changing along the car. It was shown by Keogh et Al. that as the path curvature increases the difference between drag along the curvature and the force coefficient along a straight axis increases.

Due to the changing velocity from the inside to the outside of the corner, choosing the correct velocity for force coefficient calculation becomes a dilemma. Two options can be considered for the choice of velocity. Either a constant velocity at the center of the freestream or using a distributed velocity. In another work by Keogh et Al. [6] it was shown that the choice of velocity only makes a 0.05% difference in the drag coefficient.

2 Methodology

2.1 Geometry

2.1.1 Car model and computational domain

The initial car model CAD was developed and 3D printed by Edouard Buri [7] however since the model was designed for experimental purposes, it had to be simplified and adapted for CFD simulations. The first step was to convert the body work from shell to a solid body. The latter was done using the ANSYS SpaceClaim fill tool. First, the body was divided to smaller entities and afterwards each entity was converted to solid. Then the parts were attached to make up the chassis structure. The same process was followed for the front wing and parts of the rear wing. Figure 2.1 compares the initial and final bodywork of the model. It can be observed that the final structure has a significantly lower number of faces specially around the halo and top air intake areas. This was done in order to eliminate small faces which could hinder the mesh quality.

The computational domain was created according to the existing tunnel at the laboratory. The tunnel has a height of 740 *mm* and width of 1500 *mm*. The inner and outer arc radius are 3000 *mm* and 4500 *mm* respectively. The curvature angle is 24°. The center of the corner was placed at (0, 0, -3000) *mm*. Figure 2.2 presents the placement of the car within the tunnel. The domain with the car is shown in figure 2.3. The distance from the nose of the car to the inlet is 306.3 *mm* and distance from the rear wing to the outlet is set to 274.5 *mm*.

In order to fully adjust the model to tests previously carried out by Buri, wheel camber and toe angles were introduced in the 3D model. Camber is the angle between the vertical axis of the wheel and the vertical axis of the car when viewed from rear or front. The latter is negative if the bottom of the wheel is further out than the top [8]. In the case of this project camber of -3.5° and -1.5° were applied to front and rear wheels respectively. Toe is the angle between the longitudinal axis of the wheel and the car when viewed from above. The latter is positive (toe-in) if the front of the wheel is pointing towards the car

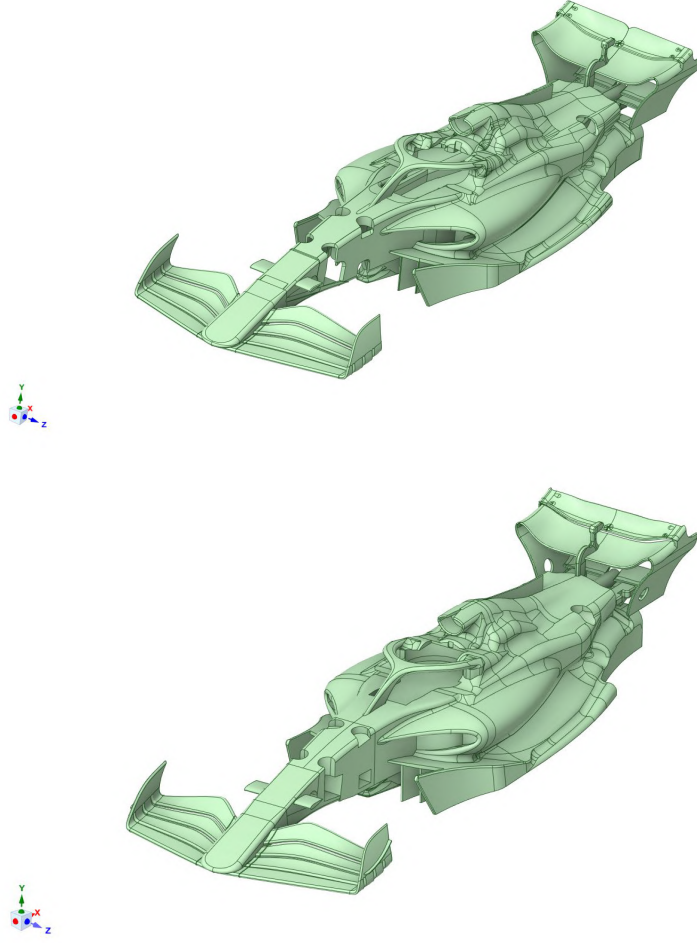


Figure 2.1: Comparison of the initial (Top) and final (Bottom) car models.

and negative (toe-out) if it is pointing away from front of the car [9]. The front toe-out and rear toe-in angles were respectively set to 1° and 1.5° .

Since the car is in a cornering state, it means that the front wheels are turning. The turning angle of the inside and outside wheels were calculated using the following formulas [10]:

$$\delta_{f,in} = \tan^{-1} \left(\frac{L}{R - \frac{T}{2}} \right) \quad (2.1)$$

$$\delta_{f,out} = \tan^{-1} \left(\frac{L}{R + \frac{T}{2}} \right) \quad (2.2)$$

Where L , R , and T are respectively the wheelbase, turn radius, and track width. Using the above formulas and using the following values for the latter variables in the same order 601.4389 mm , 3750 mm , and 1500 mm the steering angle for the inner and outer wheels are 11.33° and 7.61° respectively.

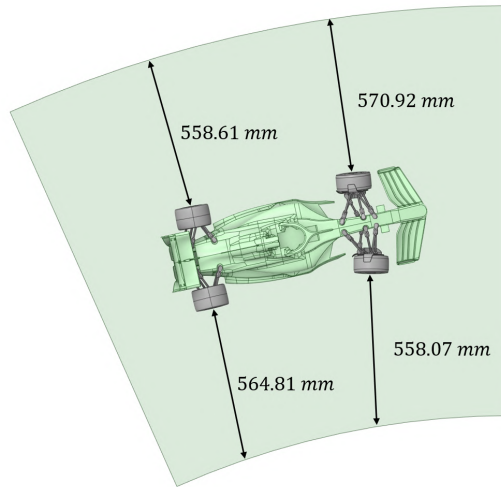


Figure 2.2: Position of the car within the tunnel. The dimensions take into account the adjustments made to the wheels.

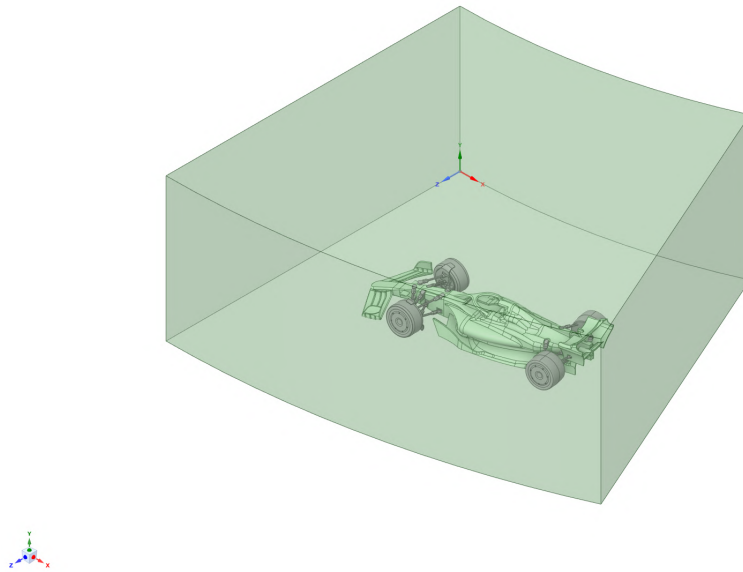


Figure 2.3: 3D representation of the computational domain.

2.1.2 Model simplifications

In order to improve the meshing process and fully adapt the model for an external aerodynamics simulation certain parts of the model were either simplified or removed. This section will provide an overview of each simplification.

2.1.2.1 Suspensions

The first step in order to simplify the model was to remove the screws connecting the suspension to the body. The latter would avoid any contact and small gaps between the elements which can cause mesh failure. Additionally, in order to better capture the airflow over these zones, the holes were filled. Figure 2.4 compares the before and after simplification of the suspension heads.

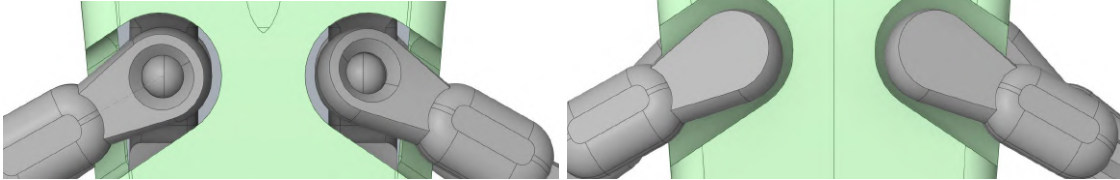


Figure 2.4: Comparison of the initial (Left) and final (Right) suspension heads.

Another important geometrical factor that affects meshing is the proximity of components to each other. If the elements are too close, the generated volume mesh will have low quality. In order to avoid this, one of the suspension bars which was placed between the rear wing and the bodywork was completely removed. Additionally, a suspension head was removed for the same reason. Figure 2.5 presents the removed rear suspension element and figure 2.6 shows the modified suspension geometry.

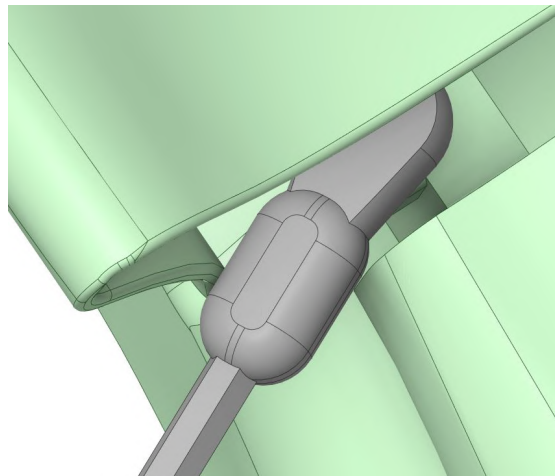


Figure 2.5: Image of the removed suspension.

2.1.2.2 Wheels

The wheels of the car were simplified by removing any unnecessary depths around and at the center of the tyres. Additionally, the inner surface of the wheels was filled to avoid any airflow entering the wheel. The latter was done despite the inside of the wheels being exposed to airflow on the actual model. However, on the physical model, the inside

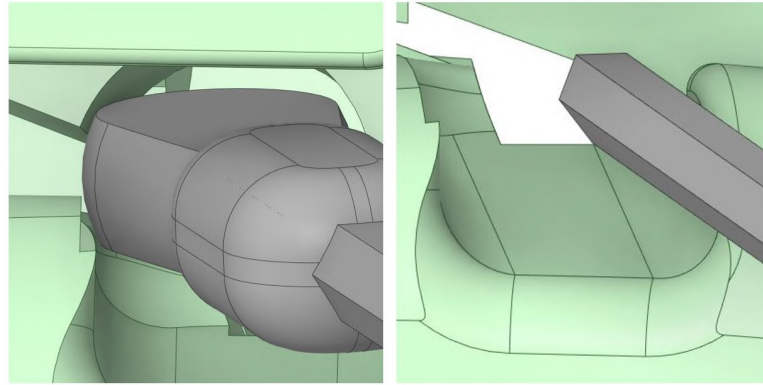


Figure 2.6: Image of the old (Left) and new (Right) rear suspension.

of each wheel is filled with the motor and its components which blocks the flow. As a result, the simplification of filling the wheels closely approximates the reality of the model. Figure 2.7 represents the initial and final state of the wheels.

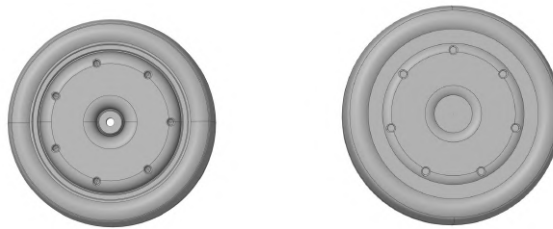


Figure 2.7: Comparison of the initial (Left) and final (Right) state of the wheels.

2.1.3 Front and rear wing

In order to avoid mesh failure at the edges of the wing elements, small fillets were added to the trailing edges of the rear wing and the beam wings. The size of the fillets for the rear wing elements is shown in figure 2.8. The sizing of the fillets was restricted by the limitations of the geometry since certain sizes did not result in desirable changes. Additionally, a fillet with the radius of 0.2 mm was added to the trailing edge of the front winglets.

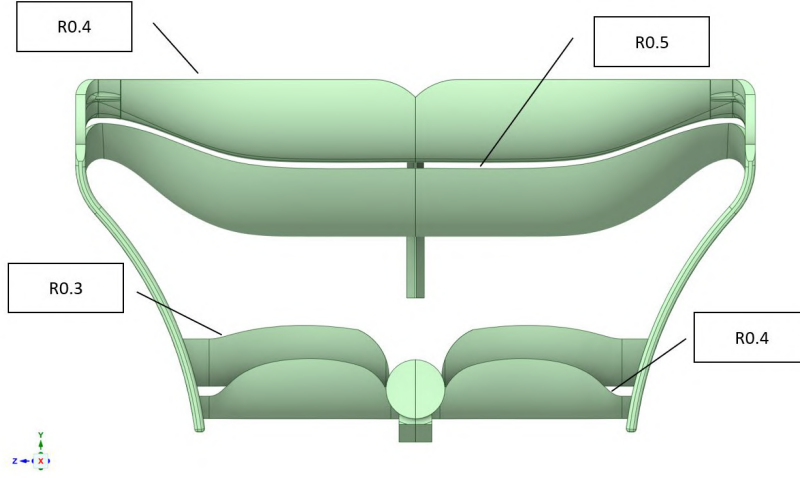


Figure 2.8: Fillet sizing of the rear wing elements.

During the meshing process it was observed that the mesh orthogonality was adversely affected by the small joints connecting the front wing elements to each other. As a result these elements were removed to improve the mesh quality. Figure 2.9 presents the small joints that were subsequently removed.

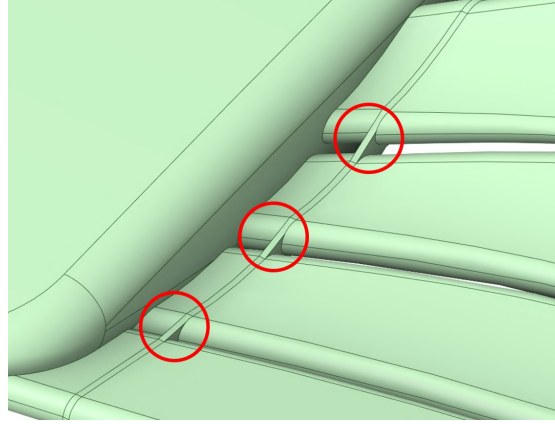


Figure 2.9: Front wing joints indicated by red circles.

2.1.4 Element names

This section will provide a visual presentation of different elements of the car and their names which will be later used as a reference throughout this report. Figure 2.10 presents the name of the visible elements of the car. Smaller elements named chassis spaces, wing spaces, winglets, and radii are presented respectively in figures 2.12, 2.11, 2.13, and 2.14. These elements are named separately in order to provide the possibility of local refinement

during the meshing process. The fillets under the wheels called radii were created to avoid any creation of sharp edges at the point of contact between the wheels and the ground. The fillets of the rear wing which are named wing fillets are presented in figure 2.8. Additionally, the naming of the computational domain is presented in figure 2.15. Furthermore, the wheel holes named selection group corresponds to the holes on the surface of the wheels which can be seen in figure 2.7.

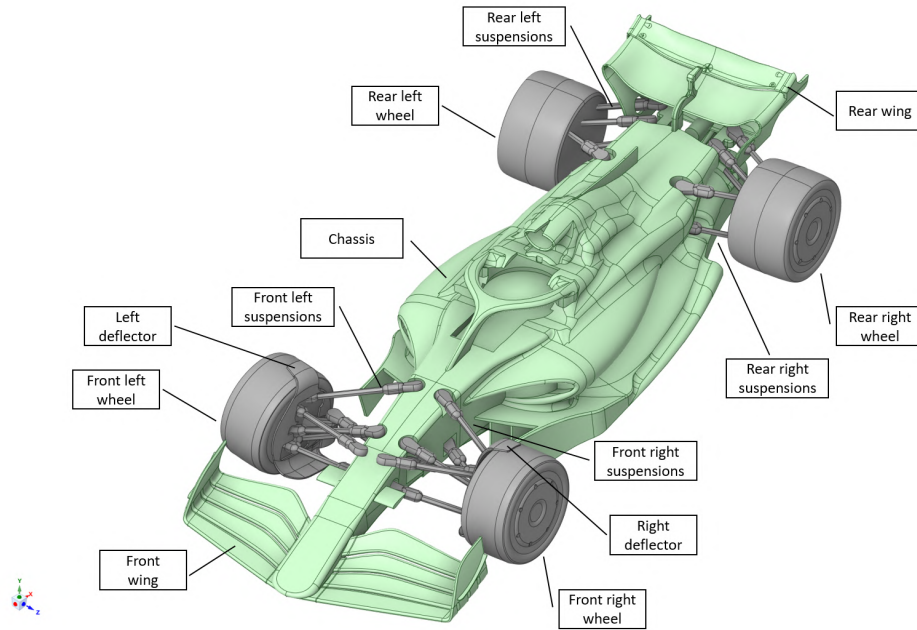


Figure 2.10: Name selection of various car elements.

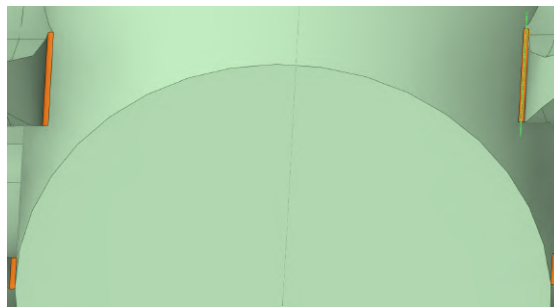


Figure 2.11: Wing spaces are four small fillets at the intersection of the beam wings and the exhaust highlighted in red.

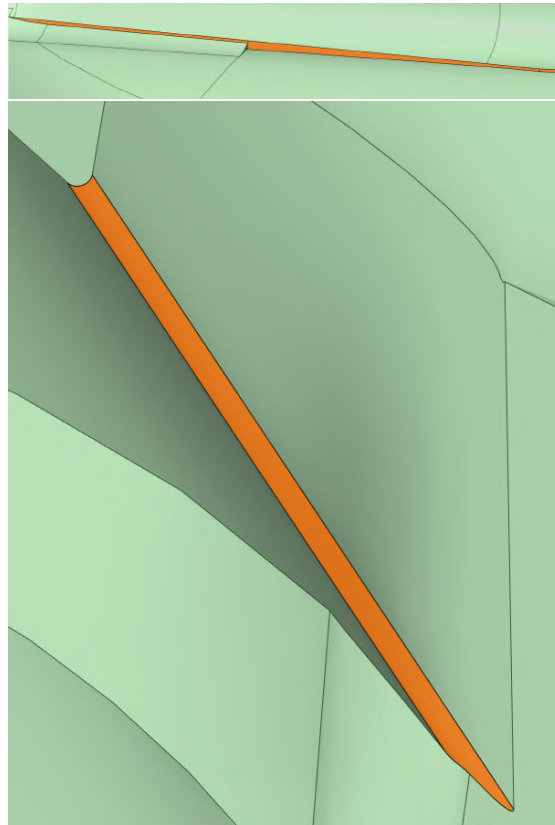


Figure 2.12: Elements named chassis spaces are small surfaces along the edge of the chassis and floor (Top) and two fillets on both sides of the exhaust (Bottom).

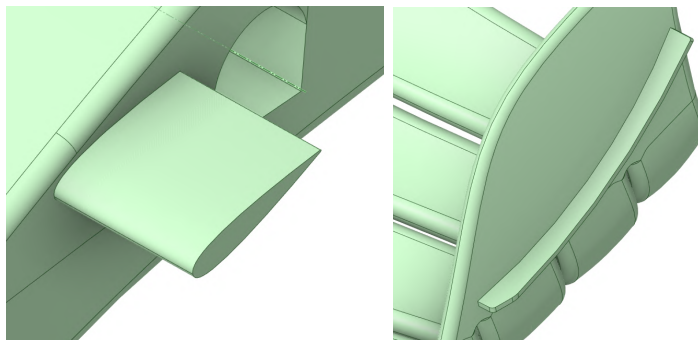


Figure 2.13: Winglets group is composed of the two larger winglets behind the front wing (Top) and two smaller winglets connected to the endplates (Bottom).

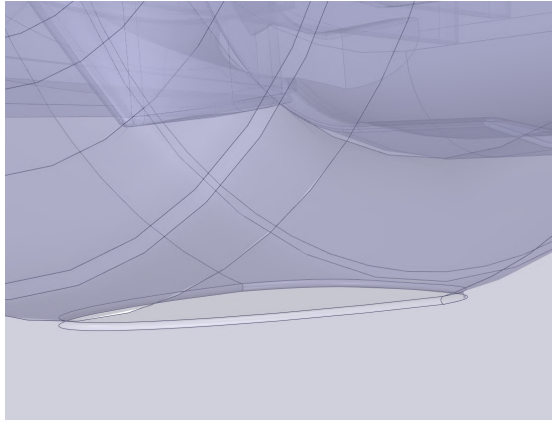


Figure 2.14: The name selection group called radii is created under each wheel.

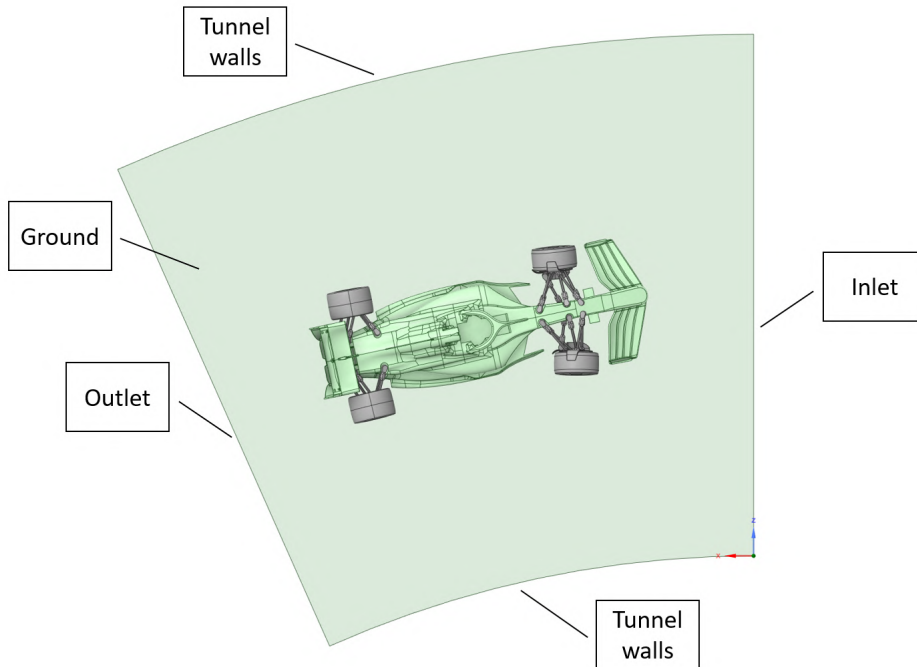


Figure 2.15: Name selection of the computational domain (viewed from above). The top surface of the tunnel is also part of the tunnel walls group.

2.2 Governing equations

2.2.1 Continuity equation

The continuity equation states that in the absence of mass source or sink terms, mass is conserved.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0 \quad (2.3)$$

Where ρ is the density and \mathbf{v} is the flow velocity vector field. In the case of the incompressible flow equation 2.3 reduces to:

$$\nabla \cdot \mathbf{v} = 0 \quad (2.4)$$

2.2.2 Momentum equation

The Navier-Stokes momentum equation for an incompressible flow is written as:

$$\frac{\partial}{\partial t}[\rho \mathbf{v}] + \nabla \cdot [\rho \mathbf{v} \mathbf{v}] = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_b \quad (2.5)$$

Where μ corresponds to dynamic viscosity, p is the pressure, and \mathbf{f}_b corresponds to body forces. It is worth noting that the left hand side of the equation corresponds to inertial forces and the terms on the right hand side of the equation correspond to pressure, viscous, and external forces.

2.2.3 Turbulence model

The nature of the flow within the tunnel and around the car is turbulent. This can be verified by calculating the Reynolds number:

$$Re = \frac{\rho_{air} V L}{\mu_{air}} = \frac{1.225 \cdot 3.3 \cdot 0.98}{1.81 \cdot 10^{-5}} = 218875.7 \quad (2.6)$$

Where L is the length of the car. It can be observed that the Reynolds number is considerably larger than the turbulent flow criteria ($Re = 4000$).

The choice of turbulence model strongly depends on the type of simulation and the field of application. One of the widely accepted methods of modelling turbulence is Reynolds-Averaged Navier-Stokes (RANS) [11]. In this method the flow variables are decomposed into time-mean and fluctuating components. Afterwards, the equations are time-averaged [11]. For example, variable $\phi(\mathbf{x}, t)$, at time t and position \mathbf{x} is decomposed into a mean value component $\bar{\phi}(\mathbf{x}, t)$ and a fluctuating component $\phi'(\mathbf{x}, t)$:

$$\phi(\mathbf{x}, t) = \bar{\phi}(\mathbf{x}, t) + \phi'(\mathbf{x}, t) \quad (2.7)$$

Following the above methodology, the governing equations for an incompressible flow [11] become:

$$\nabla \cdot [\bar{\mathbf{v}}] = 0 \quad (2.8)$$

$$\frac{\partial}{\partial t}[\rho \bar{\mathbf{v}}] + \nabla \cdot [\rho \bar{\mathbf{v}} \bar{\mathbf{v}}] = -\nabla \bar{p} + [\nabla \cdot (\bar{\boldsymbol{\tau}} - \rho \overline{\mathbf{v}' \mathbf{v}'})] + \bar{\mathbf{f}}_b \quad (2.9)$$

Various turbulence models have been derived from RANS methodology. For external flow with flow separation, the $k - \omega$ SST model is suitable and was used for this project. This model takes advantage of the strengths of both $k - \epsilon$ and $k - \omega$ models. The former

is known to perform well away from the walls while the latter is known to be a robust model with the capability of accurately calculating flows with weak adverse pressure gradient [11]. The equations for solving turbulent kinetic energy (k) and specific turbulent dissipation rate (ω) are written respectively as:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \mathbf{P} - \beta^* \rho \omega k + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right] \quad (2.10)$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{\gamma}{\nu_t} \mathbf{P} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (2.11)$$

A detailed list of parameters used in the equations can be found in appendix A.1

2.3 Computational setup

This section focuses on the computational setup of the simulations and the reason behind these choices. OpenFOAM (v2212 and v2112) was used to simulate the flow. These simulations were carried out on the work station provided by the Center for Hydro and Aero laboratory of HEPIA and EPFL's Helvetios cluster. The workstation at the laboratory is equipped with Intel(R) Xeon(R) CPU with 24 cores and 128 Gib of RAM. The Helvetios cluster contains 287 compute nodes each equipped with 2 Skylake processors running at 2.3 GHz, with 18 cores each and 192 GB of DDR3 RAM [12]. The simulations were conducted in parallel mode in order to improve performance and computational time. OpenFOAM uses the *decomposeParDict* file to obtain the settings for running parallel simulations. The settings of this file can be found in appendix A.2

2.3.1 Solver choice

OpenFOAM offers a diverse set of solvers. Since the flow around the car is slow and never reaches $Ma = 1$, the choice of the solver narrows down to incompressible solvers. Additionally, the flow is analysed in steady-state. As a result, the choice of solver narrows down to the SimpleFoam solver which uses the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm [13].

SIMPLE is a segregated solver. Therefore, it solves the governing equations sequentially and inserts the solution of the previous equation into the next. The SIMPLE algorithm first solves the momentum equation to obtain a velocity field \mathbf{v}^* . However, the latter does not satisfy the continuity equation and must be corrected. Therefore, a new equation from the momentum and continuity equation is assembled to find the pressure field p' . The latter is then used to find the correct velocity field \mathbf{v}^{**} which satisfies the continuity equation. Afterwards, the turbulence equations are solved. This process is repeated until convergence is reached [14]. Figure 2.16 presents a simplified flow chart of this process.

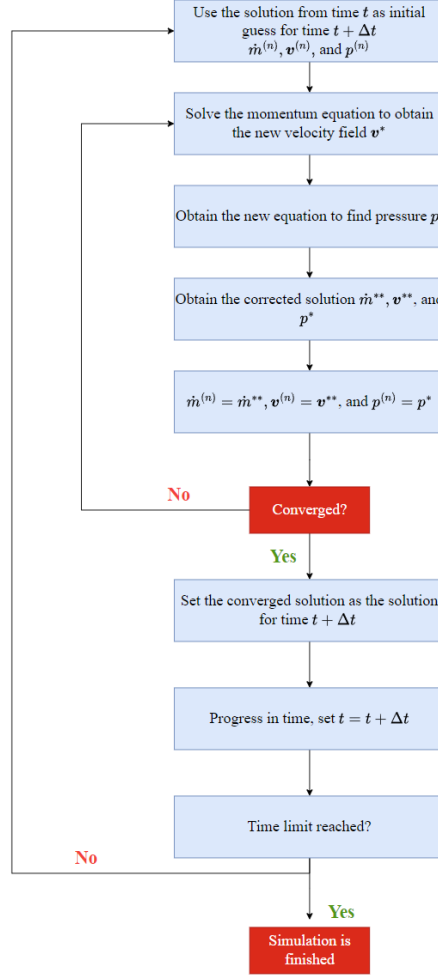


Figure 2.16: SIMPLE algorithm flow chart.

2.3.2 Boundary conditions

The boundary conditions in OpenFOAM workflow are setup within the 0 folder. The latter contains the files of the simulation variables U , p , ω , and k . It also contains the wall function file called nut. The boundary conditions of the simulation are setup within each file for all the model elements. Detailed setup of the boundary conditions can be found in appendix [A.4](#).

2.3.2.1 U and p boundary conditions

For the velocity field the following conditions were imposed:

- The inlet boundary condition in principle is a velocity inlet. The latter is normally defined by the *fixedValue* condition which sets the velocity of the flow to a fixed

value at the inlet. However, in reality at the inlet of the tunnel a linear velocity gradient along the corner radius is imposed where the velocity on the far corner is higher than the near corner. The slope of this gradient can be characterized as:

$$\omega = \frac{v}{r} = \frac{3.3}{3.75} = 0.88 \text{ rad/s} \quad (2.12)$$

Where r and v correspond to the radius and velocity at the center of the inlet respectively. Since we know that at $z = 0.75 \text{ m}$ the velocity is 3.3 m/s the following equation is obtained:

$$v = 0.88z + 2.64 \quad (2.13)$$

The latter was implemented in OpenFOAM using the **groovyBC** library which allows the user to implement custom boundary conditions.

Listing 2.1: Inlet boundary condition

```

1      inlet
2      {
3          type                groovyBC;
4          variables
5          (
6              "zp2=pos().z;"    //z coordinate
7
8              "w=0.88;"
9              "U_tx=w*zp2+2.64;"
10             "U_tz=0;"
11             "U_ax=0;"
12         );
13         valueExpression "vector(U_tx, U_ax, U_tz)";
14         value uniform (0 0 0);
15     }
```

- Since the outlet boundary condition is a pressure outlet, in the U file the condition for this patch is defined as **zeroGradient**. The latter applies a zero gradient condition from the internal field to the patch face.[15]
- On the experimental setup, the car is placed on a rotating belt. Therefore, the ground and the wheels are in rotation. In order to replicate this condition in the simulations the **rotatingWallVelocity** boundary condition was used. The ground rotates at the same angular velocity of the free stream air, 0.88 rad/s . The wheels on the inside and outside of the corner rotate at an angular velocity of 51.59 rad/s and 55.72 rad/s respectively.
- The remaining elements were defined as **noSlip** walls reflecting the actual boundary inside the tunnel.

For the pressure field the following conditions were imposed:

- Since the outlet boundary condition is set to pressure outlet and the flow is exposed

to atmospheric pressure, the *fixedValue* boundary condition with a value of zero was used.

- For the remaining elements the *zeroGradient* condition was used.

2.3.2.2 ω , k , and nut conditions

The turbulence initial conditions and wall functions are defined within the ω , k , and nut files. Since the $k - \omega$ SST turbulence model is being used there is no need for a wall function. There are two ways to implement the latter in OpenFOAM. The first method is to only use a wall function for ω . In the other method, no wall functions are used for ω , instead the specific turbulence dissipation term has a fixed value at the wall based on Menter's boundary condition at the viscous sub-layer [16]. As a result the following conditions were applied:

- For ω , all the walls are set to use the *omegaWallFunction*. This condition uses two different formulas to obtain the specific turbulent dissipation in the viscous (ω_{vis}) and logarithmic (ω_{log}) sub-layers.[17]

$$\omega_{vis} = \frac{6\nu_w}{\beta_1 y^2} \quad (2.14)$$

Where ν_w is the kinematic viscosity of the fluid near the wall, y is the wall-normal distance, and $\beta_1 = 0.075$.

$$\omega_{log} = \frac{\sqrt{k}}{C_\mu \kappa y} \quad (2.15)$$

Where k is the turbulent kinetic energy, C_μ is the empirical model constant, and κ is the von Karman constant.

The initial value of ω was set to 0.01 1/s. It is worth noting that this value does not affect the final solution since it is only an initial condition. It was observed that smaller values of ω improve the stability of the simulation at the start while larger values led to divergence.

For the inlet the *fixedValue* condition was used. At the outlet the *inletOutlet* condition was used. The latter is similar to *zeroGradient* with the exception that in case of back flow a specific value defined by *inletValue* is assigned to the returning flow.

- In Menter's work the boundary condition for turbulent kinetic energy is set to zero [3]. In this work the *fixedValue* condition with a value of 10^{-10} was used.
- Since no wall functions were used, in the nut file the *calculated* condition was assigned to all the boundaries. The latter implies that the boundary is not evaluated and only a specific user-defined value is assigned. In this case this value was set to zero. It is important to note that due to a bug on OpenFOAM v2212, the

calculated condition could only be applied to inlet and outlet. As a result, the *nutLowReWallFunction* was used as a replacement solution which provided the same results by setting turbulent viscosity (ν_t) to zero.[18][19]

2.3.3 Solver schemes

The numerical schemes used for the simulation which have an important impact on the stability and accuracy of the solution are defined in the fvSchemes file found in appendix A.5.

2.3.3.1 Gradient schemes

In OpenFOAM the gradient scheme is defined by the following format [20]:

$$\text{grad}(U) < \text{optionallimiter} > < \text{gradientscheme} > < \text{interpolationscheme} >$$

The most common gradient scheme for OpenFOAM simulations is the Gauss scheme. The latter uses the Gauss theorem to calculate the cell gradient:

$$\int_V (\nabla \cdot \mathbf{u}) dV = \oint_S (\mathbf{n} \cdot \mathbf{u}) dS \quad (2.16)$$

This method requires interpolation of values from cell centers to face centers. As a result, an interpolation method must be used. In this case the linear interpolation method is used.

To avoid large values of turbulence parameters ω and k , gradient limiters were imposed. OpenFOAM offers two general types of limiters, cell and face. Face limiters are more dissipative but provide more stability [21]. Therefore, face limiters were used in this work.

2.3.3.2 Divergence schemes

The divergence scheme for the property Q under the influence of the velocity field ϕ is defined as [22]:

$$\text{div}(\phi, Q) \text{ Gauss } < \text{interpolationscheme} > \text{ grad}(Q);$$

The divergence schemes are responsible for treating the divergence terms except the Laplacian terms which are treated by their own scheme. Divergence terms can be divided into two types, advective terms such as $\nabla \cdot (k\mathbf{U})$ and diffusive terms such as $\nabla \cdot \nu(\nabla \mathbf{U})^T$ [23]. In general the non-advective terms use the Gauss linear scheme. Due to the complexity of advection terms different interpolation schemes are used to solve them. OpenFOAM offers various first and second order schemes. While first order schemes are more stable and in general yield better residuals, they lack accuracy [24][4]. Therefore,

in this work second order schemes were used. The most commonly used second order schemes in OpenFOAM are linear, linearUpwind, LUST, and limitedLinear. In this work, the linearUpwind scheme was used. The latter relies on second order upwind differencing scheme to evaluate face value of the cell. This method uses the following formulation:

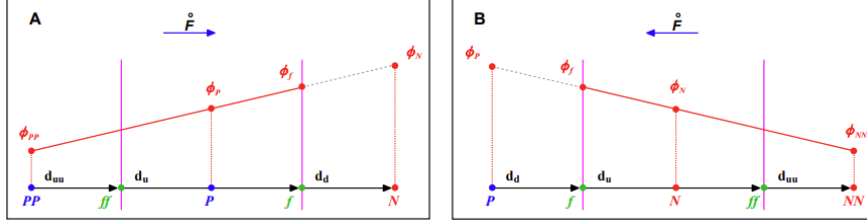


Figure 2.17: Second order upwind differencing scheme setup in the case of mass flow-in (A) and flow-out (B).[4]

$$\phi_f = \begin{cases} \frac{3}{2}\phi_P - \frac{1}{2}\phi_{PP} & F^\circ \geq 0 \\ \frac{3}{2}\phi_N - \frac{1}{2}\phi_{NN} & F^\circ < 0 \end{cases} \quad (2.17)$$

Where ϕ is a quantity in the simulation.

In the above form the upwind scheme is unbounded and can cause oscillations [4]. Therefore, gradient limiters (ψ) are added to prevent unstable behaviour. As a result equation 2.17 becomes:

$$\phi_f = \begin{cases} \phi_P + \frac{1}{2}\psi_P^-(\phi_P - \phi_{PP}) & F^\circ \geq 0 \\ \phi_N + \frac{1}{2}\psi_P^+(\phi_N - \phi_{NN}) & F^\circ < 0 \end{cases} \quad (2.18)$$

2.3.3.3 Surface normal gradient and Laplacian schemes

In the fvSolution file, *snGradScheme* and *LaplacienScheme* refer to the schemes that are used to treat surface normal gradient and Laplacian terms respectively. The choice of scheme for both of these discretization is the often the same [4], therefore they will be discussed together. They are defined in the following format [25, 26]:

$$snGrad(Q) < snGradscheme >$$

$$laplacian(gamma, phi) Gauss < interpolationscheme > < snGradscheme >$$

In order to evaluate Laplacian terms, surface normal gradients are required [23]. In principle, this calculation is second order accurate for an ideal gradient which is normal to the face [23]. However, this is rarely the case in most meshes and as a result a correction has to be implemented. In cases where the solution is not stable enough an additional

limited scheme represented with ψ has to be applied where [23]:

$$\psi = \begin{cases} 0 & \text{corresponds to } \textbf{uncorrected} \\ 0.333 & \text{non-orthogonal correction} \leq 0.5 \times \text{orthogonal part} \\ 0.5 & \text{non-orthogonal correction} \leq \text{orthogonal part} \\ 1 & \text{corresponds to } \textbf{corrected} \end{cases}$$

In general, **corrected** scheme is used for meshes with a non-orthogonality of less than 75° [4]. The non-orthogonality of meshes in this work were less than the mentioned value as a result the **corrected** scheme was used.

2.3.3.4 Interpolation

The interpolation scheme treats the interpolation of values from cell centers to face centers [23]. In OpenFOAM the most commonly used scheme is the linear interpolation scheme as it was the case in this work.

2.3.4 Solution schemes

The **fvSolutions** file contains the settings and tunings on how the equations and matrices must be solved and under which conditions the simulation should stop. Similar to solver schemes, the choices in this file can have an important effect on the duration and accuracy of the simulation. The detailed **fvSolutions** file is presented in A.6.

2.3.4.1 Solvers

The first section of the **fvSolutions** contains the solver setting for each of the simulation variables. The components of this section are:

- **solver**: Corresponds to the type of linear solver that OpenFOAM uses to solve the matrix equations [27]. The geometric-algebraic multi-grid (GAMG) solver is known to perform well for symmetric matrices. Additionally, the **smoothSolver** with **GaussSeidel** smoother is an ideal option for non-symmetric matrices.[4]
- **smoother**: Depending on the type of solver, a smoother or pre-conditioner has to be selected. The **smoothSolver** uses smoother to partially remove the error [28]. The **GaussSeidel** and **symGaussSeidel** are the two most commonly used smoothers in OpenFOAM tutorials [27].
- **tolerance**: After each iteration the residual is calculated and compared to the tolerance, the solver completely stops once the residual is below the value of this parameter.[4]

- ***relTol***: During each solver iteration the ratio between the current and the initial residual is evaluated. The current iteration of the solver stops when this ratio falls below the threshold defined by ***relTol***. [4]

The solver parameters used for this work are presented in table 2.1.

Table 2.1: Solver settings for simulation parameters.

	solver	smoother	tolerance*	relTol
p	GAMG	GaussSeidel	10^{-6}	0.01
U	smoothSolver	GaussSeidel	10^{-10}	0.1
k	smoothSolver	GaussSeidel	10^{-20}	0.1
ω	smoothSolver	GaussSeidel	10^{-20}	0.1

*It should be noted that high values of tolerance for certain parameters were set in order to ensure that the calculations for each parameter does not stop before the other. The main residual control is implemented further in the ***fvSolutions***.

2.3.4.2 SIMPLE settings

The second part of the ***fvSolutions*** file corresponds to general settings for the SIMPLE solver such as:

- ***nNonOrthogonalCorrectors***: Another way to deal with mesh non-orthogonality and improve simulation stability is the use of non-orthogonal correctors. This option enables the simulation to solve the pressure equation more than once. By default this value is set to zero meaning the pressure equation is only solved once. In general, it is recommended to set this value to at least 1 [4]. For this project the corrector was set to 2.
- ***consistency***: This option enables the use of SIMPLEC algorithm which can help relax the pressure and improve stability. [4]
- ***residualControl***: corresponds to the residual threshold for all simulation parameters. Once the residual of every parameter falls below the defined value the simulation is deemed to converge and stops. This value was set to 10^{-3} for this project.

2.3.4.3 Under-relaxation factors

Under-relaxation factors (URFs) are used to improve the stability of the simulations. The under-relaxation factor determines how much the solution changes from one iteration to the next [27]. Under-relaxation factor α varies between zero and one. Higher values

can accelerate the simulation at the cost of stability. In this work, at the start of the simulation the URF for U , k , and ω was set to 0.7 while this value for pressure was 0.3. In order to improve residual values and convergence the URF of pressure was gradually increased to 0.7 while this value for velocity was decreased to 0.3.

2.4 Mesh

The meshing process was carried out using ANSYS Fluent Meshing software. One of the benefits of the latter is the possibility of creating a poly-hex core (Mosaic) mesh which was used for this work. Mosaic meshing has the capability of generating a hex dominant mesh at a higher speed with reduced face count and improved solve time [29]. This technique generates a volume mesh with hex elements inside the volume and polyhedral elements on the boundaries. Additionally, hex dominant meshes can capture gradients better than tetrahedral meshes [5].

2.4.1 Mesh convergence study

In order to obtain an ideal mesh and to avoid using unnecessary computational cost a grid independence study was carried out. The latter was done using 5 case studies. The goal was to refine the mesh of the car and its wake. In order to achieve this two bodies of influence (BOI) were created around the car. The first body of influence was focused on the car itself while the second one covered the front and wake of the car. Figure 2.18 shows the two bodies of influence.

Three types of mesh sizing were used:

- **Body of influence:** This type allows the user to define BOIs and select a specific mesh size for the regions covered by BOI.
- **face sizing:** In this type, the selected size of the mesh elements is uniform across the whole object. This type only allows one user input which is the final size. In this work, face sizing was only used for small elements such as fillets in order to make sure the quality of mesh in these regions is adequate.
- **Curvature:** This sizing allows the user to define a minimum and maximum sizing range. This helps avoid unnecessary amount of elements where they are not needed. Additionally, a curvature angle could also be defined by using this method. Smaller curvature angles allow better meshing of highly curved surfaces. This sizing method was used extensively in this work, specially on larger parts of the car such as the chassis and wings.

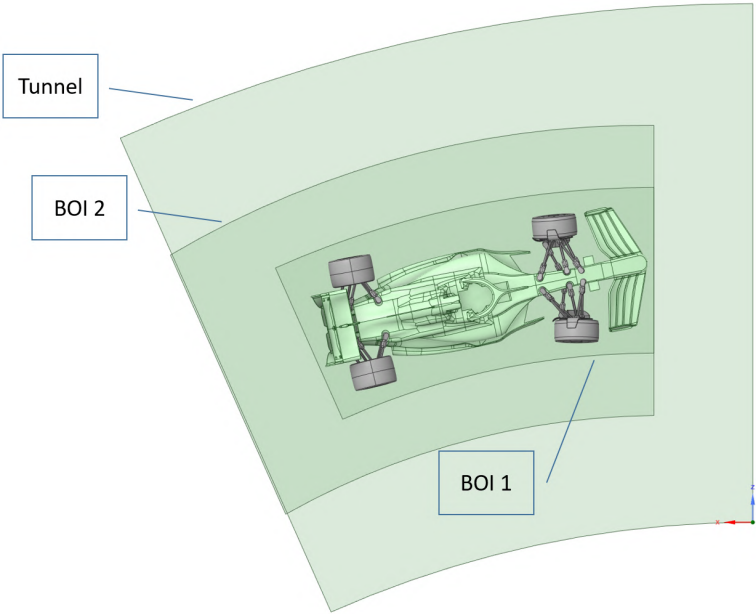


Figure 2.18: Bodies of influence placement within the tunnel.

Figure 2.19 presents the user input interface for three methods used in this work.

Name	boi_1
Growth Rate	1.2
Size Control Type	Body Of Influence
Target Mesh Size	0.006
Select By	label
Name	facesize_1
Growth Rate	1.2
Size Control Type	Face Size
Target Mesh Size	0.0001
Select By	label
Name	curvature_1
Growth Rate	1.2
Size Control Type	Curvature
Local Min Size	0.0001
Max Size	0.0008
Curvature Normal Angle	12
Scope To	faces
Select By	label

Figure 2.19: User input interface for body of influence sizing (Top), face sizing (middle), and curvature sizing (Bottom).

It should be noted that the growth rate, curvature normal angle, and scoping options were the same and according to the figure above for all the meshed parts. Detailed sizing information can be found in tables 2.2, 2.3, and 2.4.

Table 2.2: Mesh sizing of the car and bodies of influence for mesh 1 and mesh 2.

		Mesh 1		Mesh 2	
Part	Sizing method	Min [mm]	Max [mm]	Min [mm]	Max [mm]
BOI car	BOI	-	6	-	6
BOI wake		-	28	-	25
Winglets	Curvature	0.1	0.8	0.1	0.6
Deflectors		0.1	1.5	0.1	1.5
Suspensions		1	3	0.8	2
Wheels		1	3	0.8	2
Front wing		1	3	1	2
Rear wing		0.5	3	0.5	2
Chassis		0.1	2	0.1	2
Radii	Face sizing	-	0.1	-	0.1
Wing fillets		-	0.2	-	0.2
Wing spaces		-	0.1	-	0.1
Chassis spaces		-	0.1	-	0.1
Wheel holes		-	0.5	-	0.5
Total elements [Million]		11.04		11.46	

Table 2.3: Mesh sizing of the car and bodies of influence for mesh 3 and mesh 4.

		Mesh 3		Mesh 4	
Part	Sizing method	Min [mm]	Max [mm]	Min [mm]	Max [mm]
BOI car	BOI	-	6	-	6
BOI wake		-	22	-	22
Winglets	Curvature	0.1	0.8	0.1	0.8
Deflectors		0.1	1.5	0.1	1
Suspensions		0.5	1	0.5	1
Wheels		0.5	1	0.5	1
Front wing		0.4	1	0.1	1
Rear wing		0.5	1	0.1	1
Chassis		0.1	1	0.1	1
Radii	Face sizing	-	0.1	-	0.1
Wing fillets		-	0.2	-	0.2
Wing spaces		-	0.1	-	0.1
Chassis spaces		-	0.1	-	0.1
Wheel holes		-	0.3	-	0.3
Total elements [Million]		18.3		20.58	

Table 2.4: Mesh sizing of the car and bodies of influence for mesh 5.

		Mesh 5	
Part	Sizing method	Min [mm]	Max [mm]
BOI car	BOI	-	6
BOI wake		-	20
Winglets	Curvature	0.1	0.5
Deflectors		0.1	1
Suspensions		0.1	1
Wheels		0.1	1
Front wing		0.1	0.5
Rear wing		0.1	0.5
Chassis		0.1	1
Radii	Face sizing	-	0.1
Wing fillets		-	0.1
Wing spaces		-	0.1
Chassis spaces		-	0.5
Wheel holes		-	0.1
Total elements [Million]		28.62	

Mesh sizing for the remaining elements such as the ground and tunnel walls is presented in figure 2.20. It should be noted that the minimum size is the same as BOI wake and therefore varies for each mesh.

The screenshot shows a mesh sizing configuration window with the following settings:

- Minimum Size: 0.028
- Maximum Size: 0.03
- Growth Rate: 1.2
- Size Functions: Curvature & Proximity
- Curvature Normal Angle: 12
- Cells Per Gap: 5
- Scope Proximity To: edges
- ☒ Draw Size Boxes
- Separate Out Boundary Zones by Angle?: No

Figure 2.20: Sizing of other mesh parts such as the ground and tunnel walls.

Figures 2.21 to 2.23 visualise the difference of surface mesh refinement between the coarsest and finest mesh.

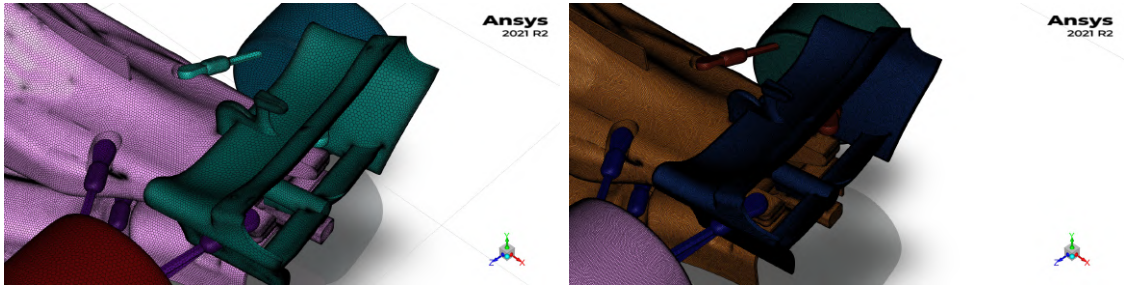


Figure 2.21: Rear wing surface mesh on the coarsest (Left) and finest (Right) mesh.



Figure 2.22: Front wing surface mesh on the coarsest (Left) and finest (Right) mesh.

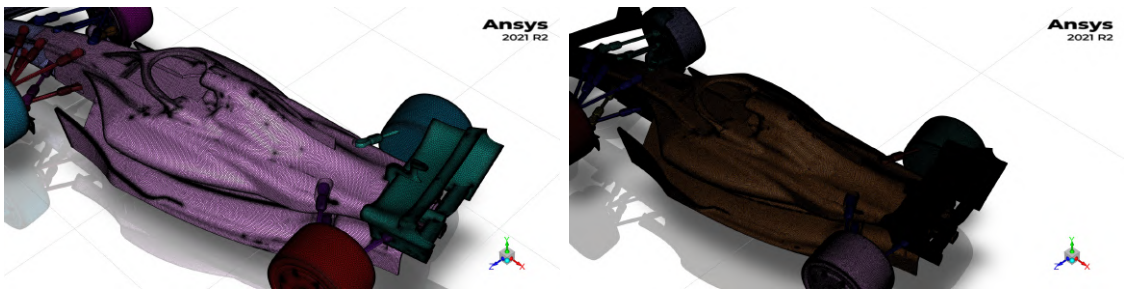


Figure 2.23: Chassis surface mesh on the coarsest (Left) and finest (Right) mesh.

2.4.2 Boundary layer (BL)

In order to accurately capture the flow near the wall, boundary layers had to be created. The exact detail of these layers can be found in table 2.5. The three types of layers that were applied are [30]:

- **Uniform:** This type of boundary layer allows the generation of layers with the same distance between them. This distance is equal to the distance between the parent node and the first layer. This method was used on the chassis and front and rear wing. Since in these parts of the car there are spaces with a very small gaps between the elements, the uniform height of the boundary layer helps avoid mesh sidestepping. The latter happens when there is not enough space to fit the entire boundary layer.[31]

- **Last ratio:** In this method, the aspect ratio of the boundary layer cells can be controlled. Unlike the uniform method, in this method the distance between the layers gradually increases. Additionally, first cell height can be adjusted. This type of boundary layer was only used for the ground and tunnel walls since there was no risk of stair stepping.
- **Aspect ratio:** This method is similar to the previous type with the only difference that no first cell height can be defined. This type of boundary was used for the remaining parts of the car.

Since most of the aerodynamic forces come from the chassis, rear wing, and front wing it was important to use 5 boundary layers to accurately capture the flow near the wall. However, this is not the case for other components. As a result, only 3 layers were used. Figure 2.24 visualises the boundary layer at the front and rear wing.

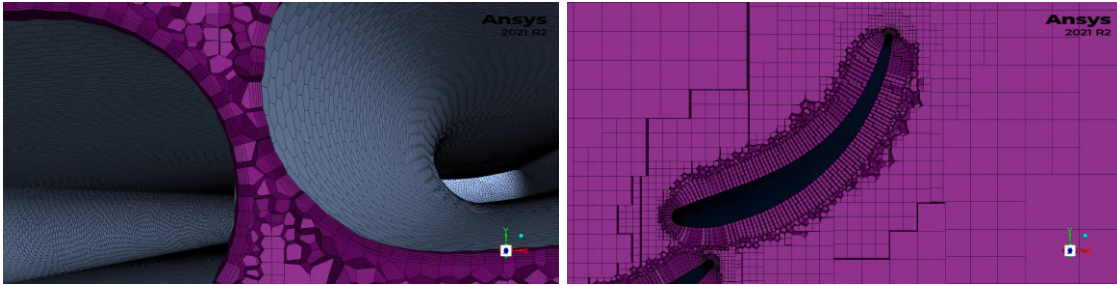


Figure 2.24: Front (Left) and rear (Right) wing boundary layer mesh.

Table 2.5: Boundary layer configuration for the car and computational domain.

Part	BL type	Layers	First layer height [mm]
Front wing	Uniform	5	1.86
Winglets			
Rear wing			
Wing fillets			
Wing space			
Chassis			2.03
Chassis spaces	Aspect Ratio	3	First Aspect ratio [-] 5
Deflectors			
Suspensions			
Wheels			
Wheel holes	Last Ratio	5	2.76
Ground		5	
Tunnel walls			

2.4.3 Grid independence criteria

In order to assess grid independence, drag and lift coefficients were used as the main criteria. These values were evaluated on the whole car with the following formulas respectively:

$$C_d = \frac{F_d}{\frac{1}{2}\rho V^2 A} \quad (2.19)$$

$$C_l = \frac{F_l}{\frac{1}{2}\rho V^2 A} \quad (2.20)$$

Where:

- F_d is the drag force.
- F_l is the lift force.
- ρ is the density of air which is equal to 1.225 kgm^{-3}
- V is the free stream velocity at the center of the tunnel equal to 3.3 m/s
- A is the frontal area of the car equal to 0.0520678 m^2

2.4.4 Mesh quality

Mesh quality is one of the most important factors that can help understand whether the mesh is suitable for simulation. The most important mesh quality criteria are:

- **Orthogonality:** This is the measure of angular deviation between the vector connecting the center of the cells (\mathbf{d}) and the vector at the center of the face (\mathbf{S}) [5]. According to OpenFOAM's documentation [23], meshes with maximum non-orthogonality of more than 85° are very unlikely to converge. The maximum acceptable value by default in OpenFOAM is set to 65.

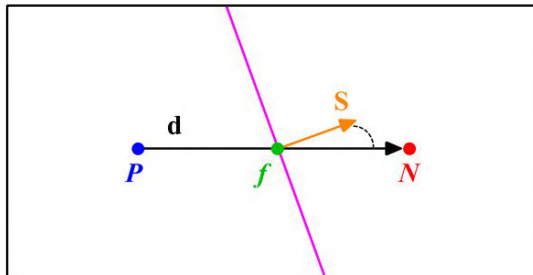


Figure 2.25: Schematic of how mesh orthogonality is defined.[5]

- **Skewness:** This metric measures the deviation between the vector that connects two cells (d) from the face center (f). This deviation is represented by Δ_i in figure 2.26. In OpenFOAM, by default the maximum value of skewness is set to 4.

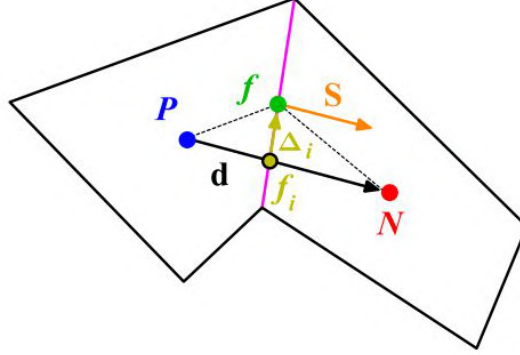


Figure 2.26: Schematic of how mesh skewness is defined.[5]

- **Aspect ratio:** This metric is the ratio between the largest (Δx) and smallest (Δy) sides of the cell.

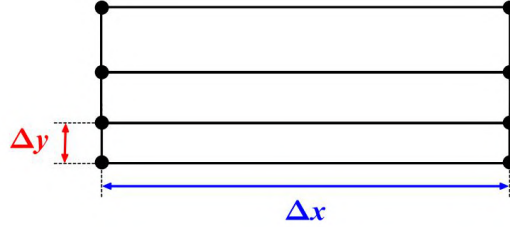


Figure 2.27: Schematic of how mesh aspect ratio is defined.[5]

The quality of meshes that were studied is listed in table 2.6. It can be observed that as the mesh is refined the quality of the mesh improves.

Table 2.6: Mesh quality of the studied meshes.

Mesh	Non-orthogonality [Avg.-Max.]	Max. skewness	Max. aspect ratio
Mesh 1	10.93-68.33	4.16	52.51
Mesh 2	10.91-66.51	4.53	54.23
Mesh 3	9.75-64.7	4.22	30.5
Mesh 4	9.47-62.39	4.26	29.6
Mesh 5	8.32-69.77	4.25	62.81

2.4.5 Results

2.4.5.1 Lift and drag coefficients

In order to establish the grid convergence study, the value of convergence criteria such as lift and drag coefficients have to become independent of the total number of elements in the mesh. Due to oscillations of the drag and lift coefficient, an average value over the last 500 iterations were calculated to obtain a more accurate understanding of the results. This behaviour will be further discussed in section 3.1.2. Table 2.7 presents the average values for each mesh. Figures 2.28 and 2.29 present the average values of drag and lift coefficients as a function of number of mesh elements respectively.

Table 2.7: Average value of lift and drag coefficients.

Mesh	#Elements [Million]	C_d	C_l
Mesh 1	11.04	0.932	-0.697
Mesh 2	11.46	0.919	-0.705
Mesh 3	18.3	0.921	-0.725
Mesh 4	20.58	0.92	-0.729
Mesh 5	28.62	0.92	-0.688

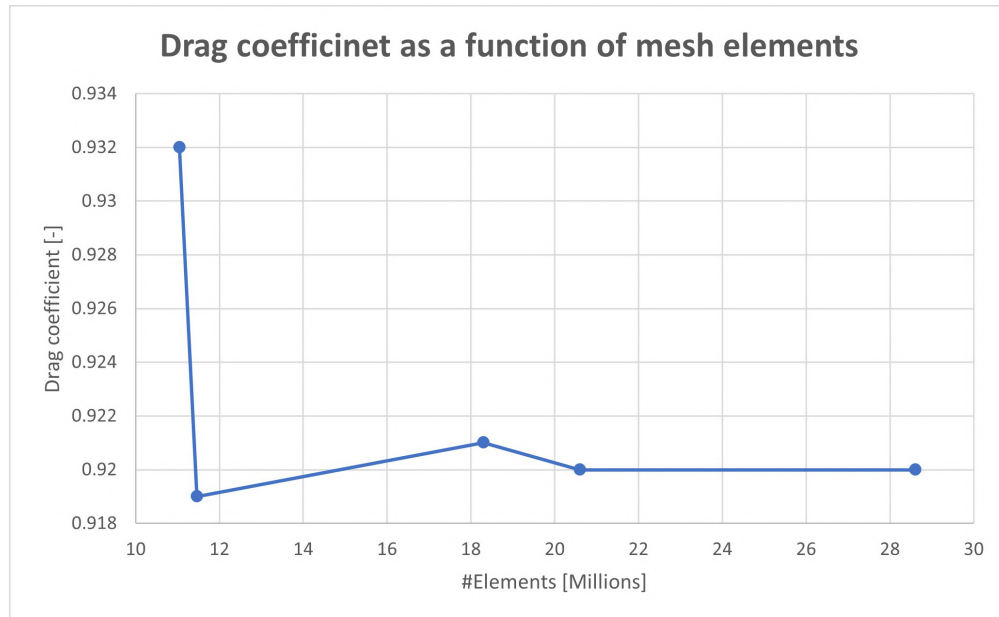


Figure 2.28: Average value of drag coefficient as a function of mesh elements.

It can be observed that the drag coefficient values reach a plateau after mesh 2. The error between the value from mesh 2 and the last value is only 0.1% which is acceptable. The average lift coefficient value shows a drastically different behaviour compared to drag. The latter is due to higher oscillations of the lift coefficient. As the mesh elements become smaller the oscillatory behaviour tends to become more intense and less stable. As a

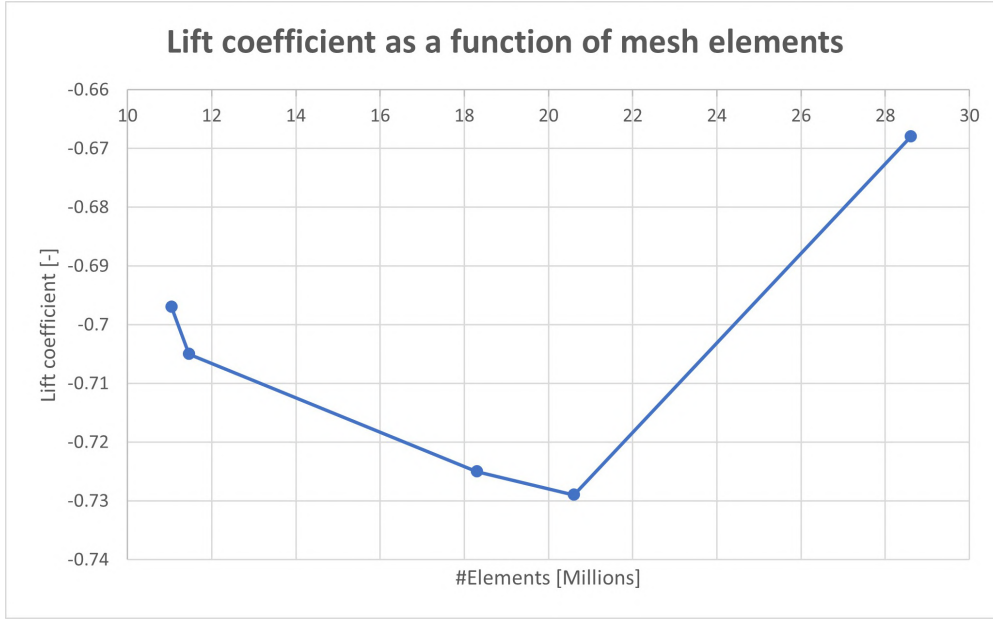


Figure 2.29: Average value of lift coefficient as a function of mesh elements.

result, no conclusive convergence can be derived from the lift coefficient. The possible causes of this behaviour and possible ways to fix it will be discussed in section 3.1.2. It is worth noting that the error between the smallest and largest value is only 9%.

2.4.5.2 Y^+ value

The Y^+ value is a measure of distance from the first grid cell to the surface [32]. This value is a good indication of how fine the mesh has to be near the boundary layers. Depending on the turbulence model and the wall treatment there can be different restrictions on this value. The recommended Y^+ range for $k - \omega$ SST turbulence model is $Y^+ < 2.5$ [33]. Since this value cannot be predicted before the simulation, during the mesh study it is monitored. Afterwards, the boundary layer mesh is adjusted accordingly to reach the required value. The Y^+ values for mesh 1 are presented in table 2.8. It can be observed that for the most important components the average value is close to the recommended threshold with the exception of tunnel walls. However, given the distance between the car and the tunnel walls and also with the emphasis on the flow around the car, this value can be accepted. Therefore, given the presented values the boundary layer was not modified. For other mesh cases the Y^+ value can be found in appendix A.7.

2.4.6 Choice of final mesh

Considering the results of the grid independence study and mesh quality metrics, the third mesh with approximately 18.3 million elements was chosen for further study of the

flow within the wind tunnel.

Table 2.8: $Y+$ values of mesh 1.

Part	Min. $Y+$	Max. $Y+$	Avg. $Y+$
Rear Wing	0.0006	16.67	2.92
Front wing	0.009	14.37	2.86
Winglets	0.002	9.39	1.93
Chassis	0.0001	15.37	2.96
Ground	0.081	17.92	3.88
Tunnel walls	4.74	18.99	13.92

3 Results and discussion

3.1 Convergence

3.1.1 Residuals

In order to evaluate the convergence of a simulation multiple criteria exist. One of the most important criteria are the residuals which measure the error between the initial and final solution at each iteration. As mentioned in section 2.3.4.2 residual threshold was set to 10^{-3} according to industry standards. In general, convergence is achieved when all the residuals fall below this threshold while reaching a plateau. Figure 3.1 presents the residual curves of the simulation.

It can be observed that except three parameters, the remaining residuals satisfy our criteria. The pressure, turbulent kinematic energy, and vertical velocity residuals don't fall below 10^{-3} . Despite the latter, they still reach a plateau and don't show any oscillatory behaviour. During the simulations it was observed that the residuals for the vertical velocity could further decrease by allowing the simulation to run for more iteration. However due to large computational time, the simulations were only conducted until 7000 iterations. Additionally, it was observed that setting the relaxation factors of turbulent kinetic energy to 0.4 would result in fulfillment of the threshold. It is also worth noting that the sudden drops in residuals are a result of URF adjustment.

3.1.2 Lift and drag coefficients

In addition to residuals, aerodynamic forces are another important factor in evaluating convergence. As mentioned before, drag and lift coefficients were used to study grid independence study. In general, it is important that these coefficients reach a plateau and settle around a value. The results for drag coefficient show that after 7000 iterations the simulation converges to a specific value with minor oscillations in the range of 0.91 – 0.92 and in the case of Mesh 4 sometimes between 0.93 and 0.94. The latter is a satisfying

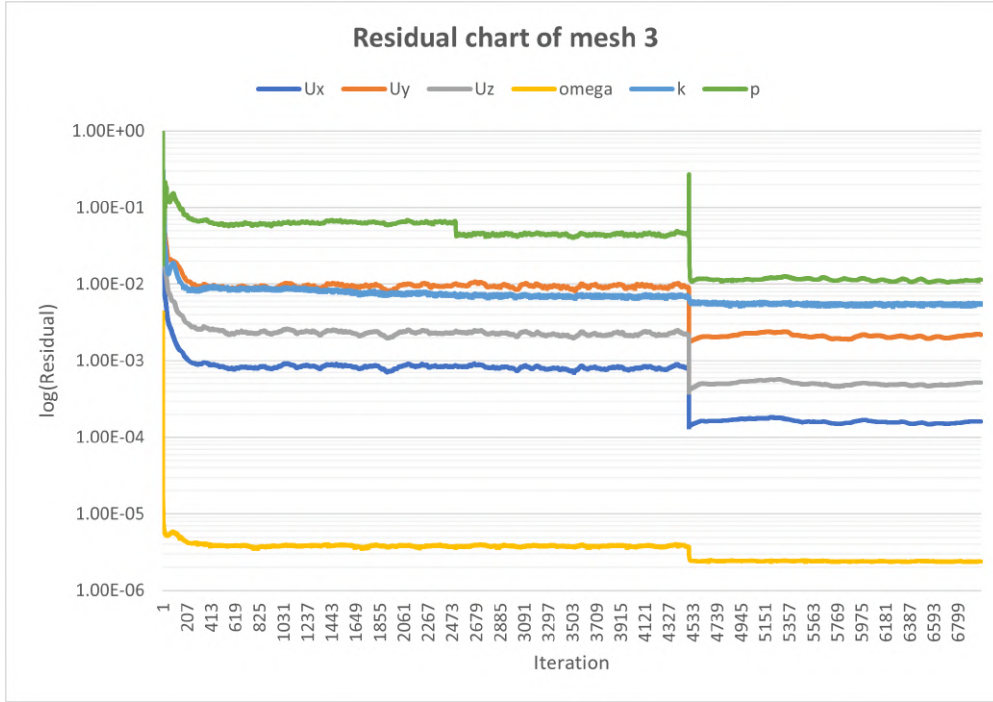


Figure 3.1: Residual curves of the final simulation. The sudden drops are a result of URF change.

outcome in terms of convergence. Figure 3.2 presents the drag coefficient curve of all the studied meshes.

In the case of lift coefficient a very different oscillating behaviour is observed. In this case oscillations are much higher and only near the end of the simulation a more relaxed behaviour is observed. However, the latter is still too high and shows that the lift coefficient has not fully converged. Figure 3.3 presents the lift coefficient curve for each mesh. The difference between maximum and minimum values of the last 500 iteration of mesh 1 and mesh 5 is 0.099 and 0.15 respectively. The latter shows that as the mesh becomes finer, the oscillations tend to increase.

The following could be the cause of such behaviour:

- Mesh quality
- Inadequate boundary condition
- Unsteady nature of the problem

As discussed in section 2.6 the mesh quality of all the cases are adequate for OpenFOAM simulations. As a result, the possibility of mesh causing oscillations is low.

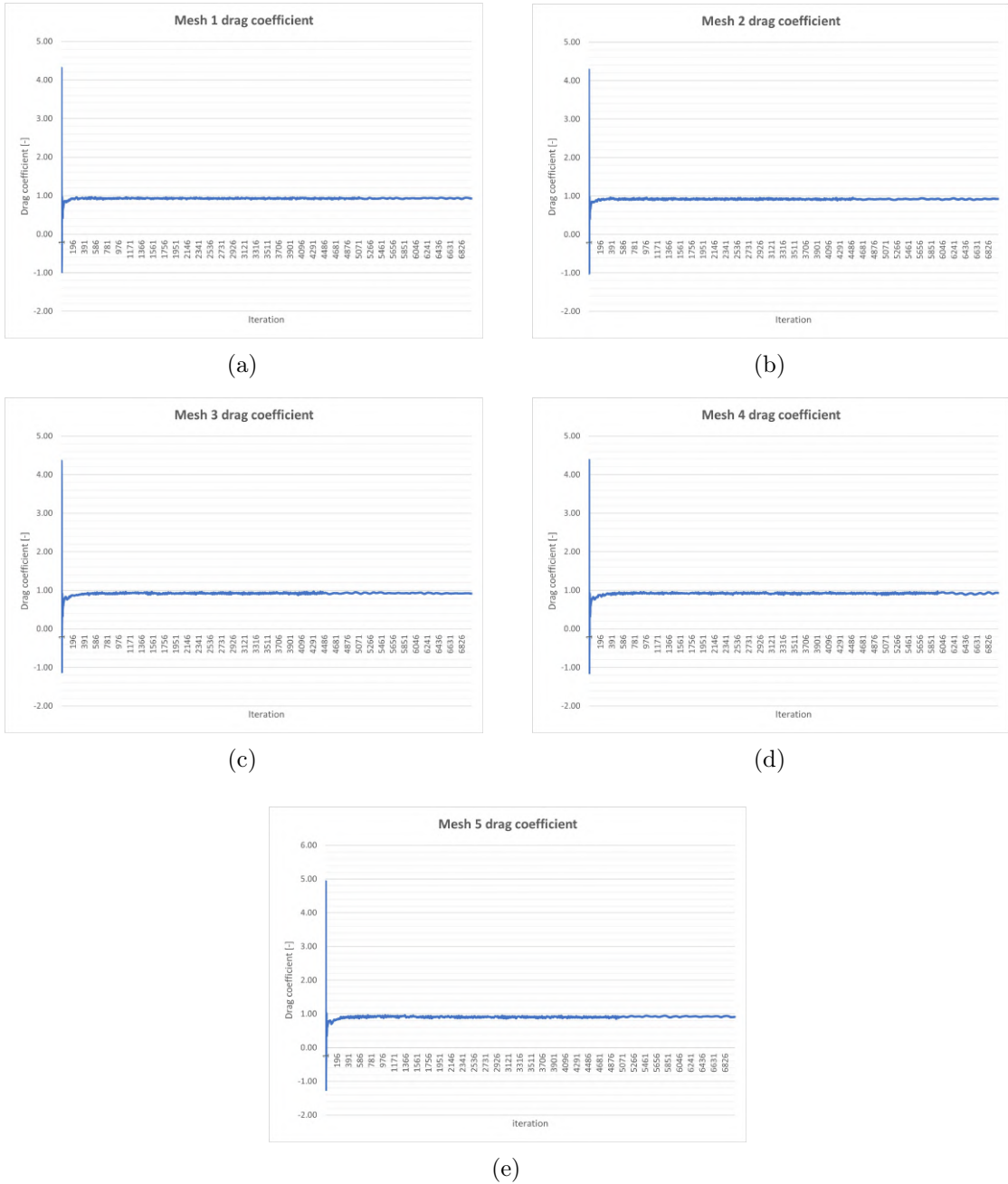


Figure 3.2: Drag coefficient curves for each mesh.

Another important factor in stability of the simulation is boundary conditions. Due to presence of a pressure gradient inside the wind tunnel, another way to implement the outlet boundary condition is to impose a pressure gradient at the outlet. In order to test this implementation, the pressure gradient at the inlet was used to characterize the outlet boundary condition. However, this caused instability and eventually divergence. As a

Chapter 3. Results and discussion

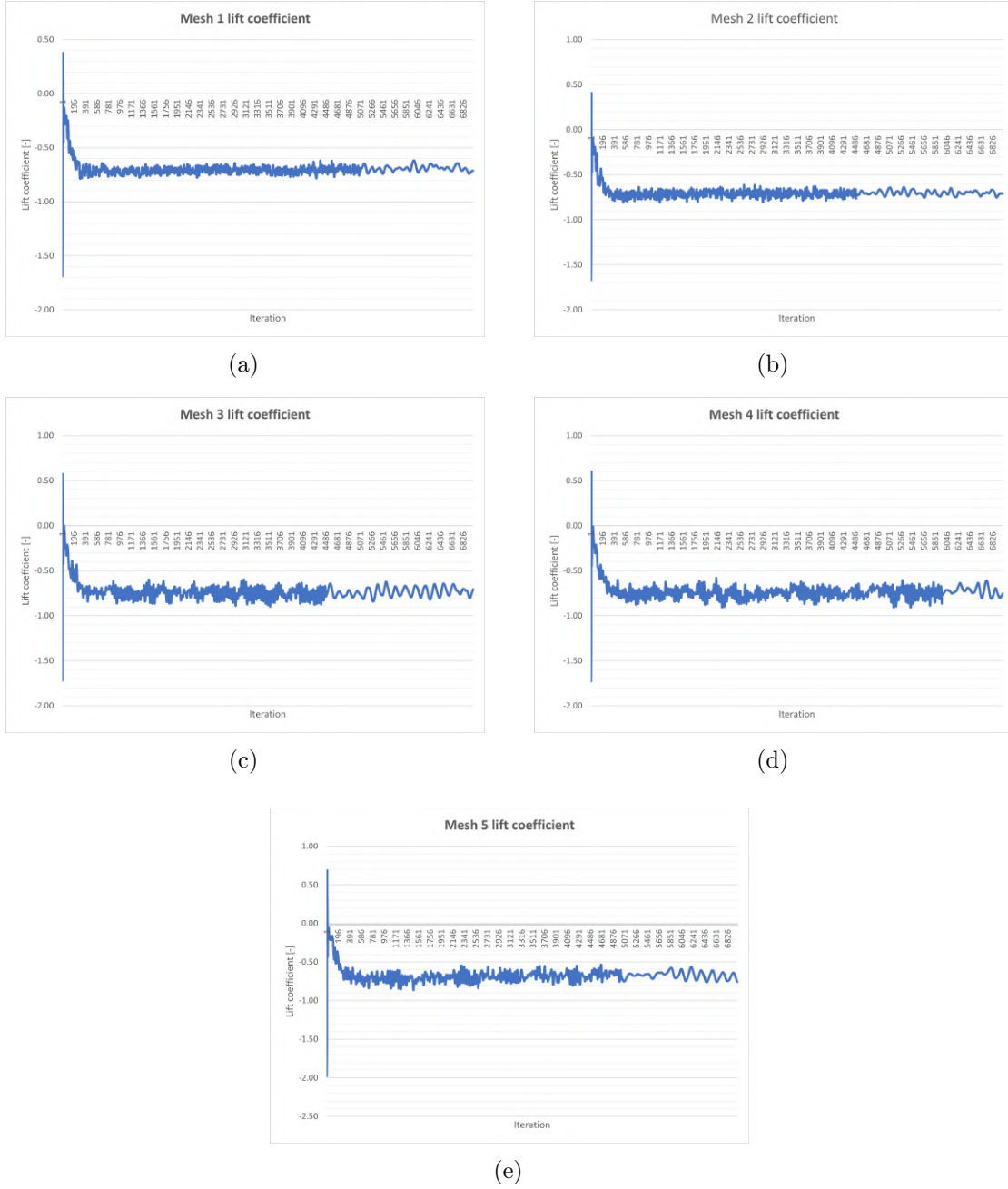


Figure 3.3: Lift coefficient curves for each mesh.

result assuming that the pressure at the outlet is equal to the atmospheric pressure is a more adequate condition as used by Keogh et Al. [34] in a curved wind tunnel.

Given the turbulent nature of the flow and the oscillatory behaviour of the lift coefficient, the flow can be unsteady. In this case, simulating an unsteady flow using steady state methods will inevitably lead to oscillation. Additionally, steady state simulation of

unsteady flows can become more unstable as the mesh is refined which is similar to the behaviour observed here. It also worth noting that the limitation of the simulation by the experimental setup could be another factor in inducing an unsteady flow. In CFD simulation it is a best practice to create a computational domain large enough to avoid influence of flow boundary condition on the flow around the car [35]. This distance is usually two times and five times the vehicle's length at the upstream and downstream of the flow respectively [35]. However, in the wind tunnel the distance between the car and the inlet is only 306.3 mm which is less than one car's length. The distance from the downstream is only 274.5 mm. Therefore, the probability of unfavorable flow conditions is higher due to such physical limitations.

3.2 Inlet and ground

Before proceeding to presentation and discussion of the final results, it would be worthwhile to ensure that the boundary conditions at the inlet and ground are correctly implemented. Figure 3.4 shows the velocity contour of the inlet. It can be observed that the velocity progressively increases as we approach the outside of the corner which is the desirable outcome. Additionally, at $z = 0$ and $z = 1.5$ the values of the boundary correspond to what is obtained with equation 2.13. A similar trend can be observed for the ground presented in figure 3.5.



Figure 3.4: Velocity contour of the inlet.

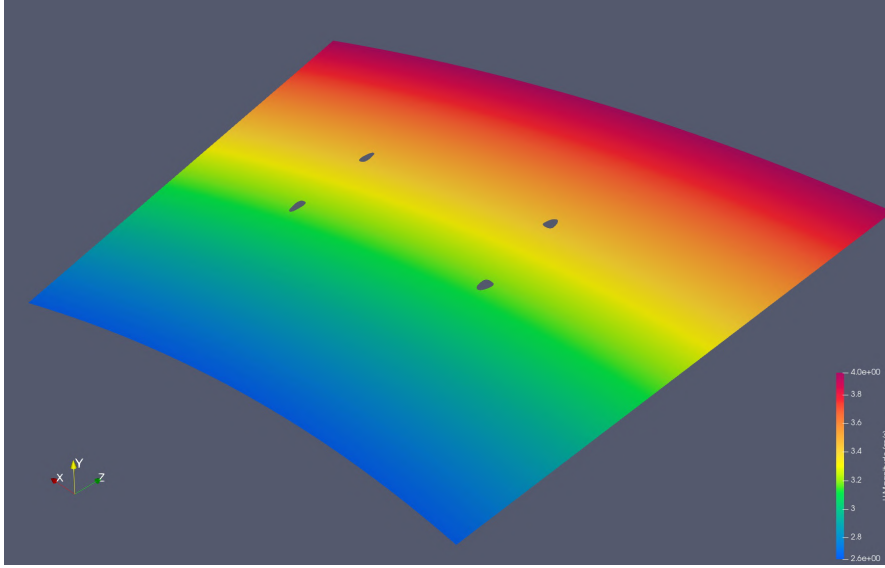


Figure 3.5: Velocity contour of the ground.

3.3 velocity contours

3.3.1 Front wing

In this section the velocity contours of various parts of the vehicle will be presented and briefly discussed. One of the most important areas of every F1 car are the wings. A considerable amount of downforce is generated by these components. Additionally, the wake of these components and their effect on racing has been subject of many studies and one of the driving forces of the new regulations proposed by the FIA. Figure 3.6 presents the velocity field behind the front wing. The difference in flow velocity between the outside and inside of the corner which is one of the most important characteristics of cornering aerodynamics can be observed in this figure. More importantly, the vortices at the trailing edge of the endplates show different velocities. This difference is approximately up to 0.5 m/s .

3.3.2 Diffuser

Figure 3.7 presents the flow on the rear bodywork and the diffuser. A distinct velocity difference can be observed in the diffuser region where the velocity on the right hand side is approximately 4 m/s while the velocity on the left side is approximately 2.5 m/s . Another interesting region is the top of the diffuser where a drop in velocity is observed. At first glance this seems to be false since the air flow under the floor is accelerated as it exits the diffuser. However, in the case of this model the diffuser surface is not smooth and the flow is blocked in certain regions. The latter is due to extruded surfaces on the floor which were created to provide housing for suspension components. This is shown

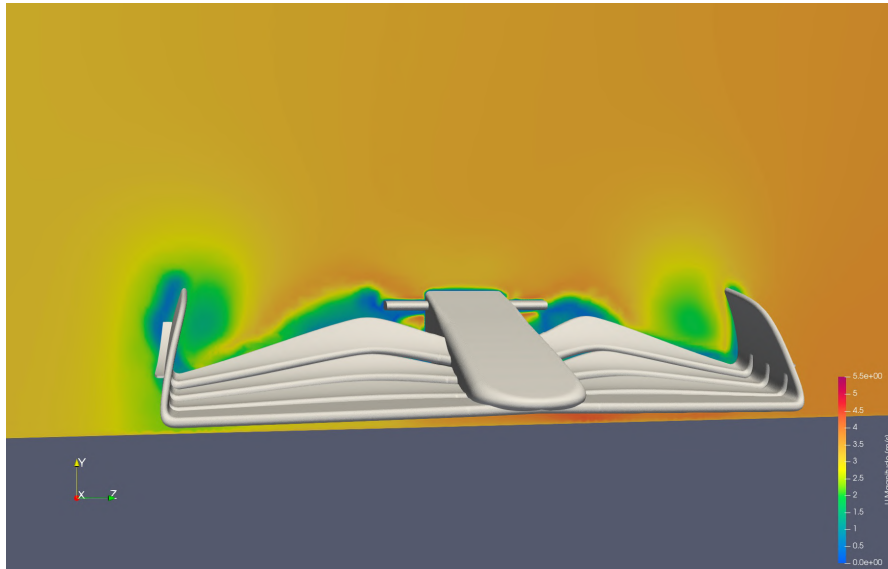


Figure 3.6: Velocity field behind the front wing.

in figure 3.8. Another interesting observation is the region between the wheels and the diffuser body. In this part we can see velocity drop on both sides and more so on the outside of the corner. This velocity drop is the result of the drag caused by suspensions and the wheel. Since the flow is much faster on the outside, the losses will also be greater as a result.



Figure 3.7: Velocity field behind the rear bodywork.

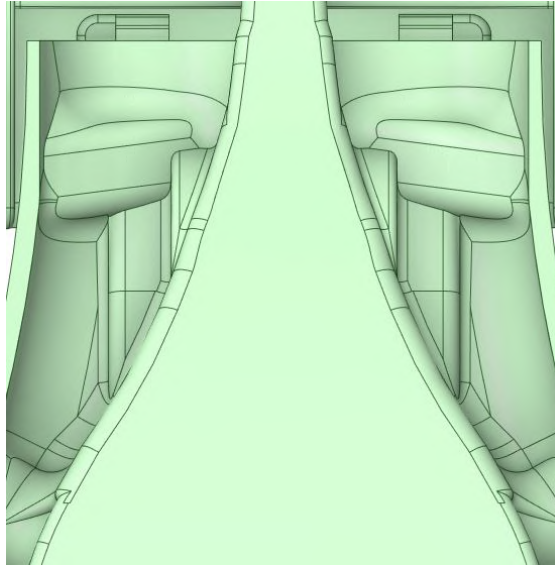


Figure 3.8: Extruded housings on the floor which cause flow blockage.

3.3.3 Rear wing

Figure 3.9 presents the wake of the rear wing. The wake is being looked at from behind the car. At the lower edges of the wake, the effect of drag from the wheels can be observed. Since the velocity on the outside is higher this effect is less pronounced on the left wheel. We can also observe that since the flow is not symmetric and there is a velocity gradient, the wake is not symmetric. An interesting region is the right lower side of the wake where a vortex with an approximate velocity of 2 m/s is formed. This is the culmination of the flow separation at front and rear wheel on the inside of the corner. Since the front wheel is turning at a sharper angle, it deflects the flow to its outside and as a result it arrives at the rear wheel with a higher velocity and is eventually separated from the rear wheel. The latter can be better observed in figure 3.10 with the help of streamlines. Additionally, we can observe that on the right side (outside of the corner) most of the flow is sucked into the area between the rear wheel and the diffuser while on the left side most of the flow is deflected to the outside of the rear wheel.

The top edges of the wake are the result of the flow over the upper wing while the lower velocity flow in the middle is the result of the flow through the beam wings, suspension components, and the wake of the wheels. Figures 3.11 and 3.12 help understand how the wake of the rear wing is created. In the first figure, the streams help visualise how the outer parts of the wake are formed while the second figure shows how the inner part of the wake is created.

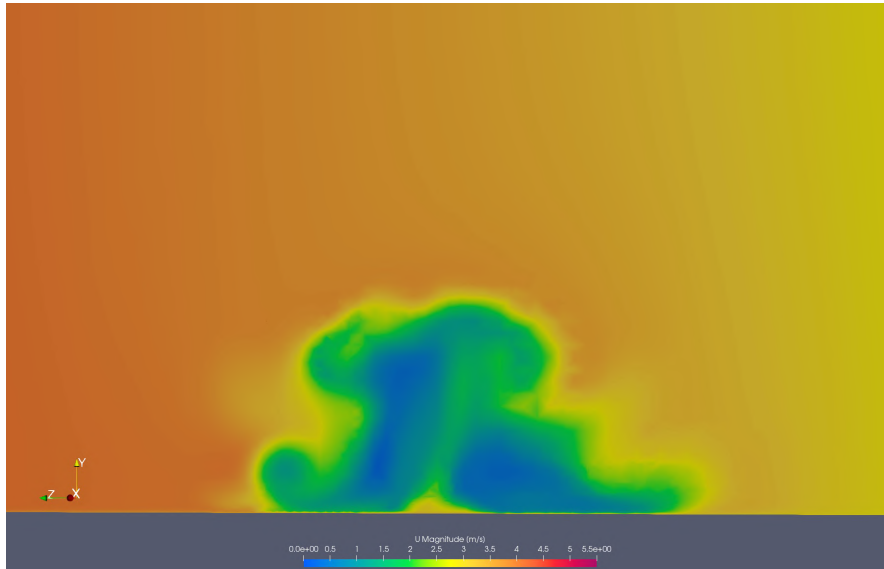


Figure 3.9: Velocity contour of the rear wing wake.

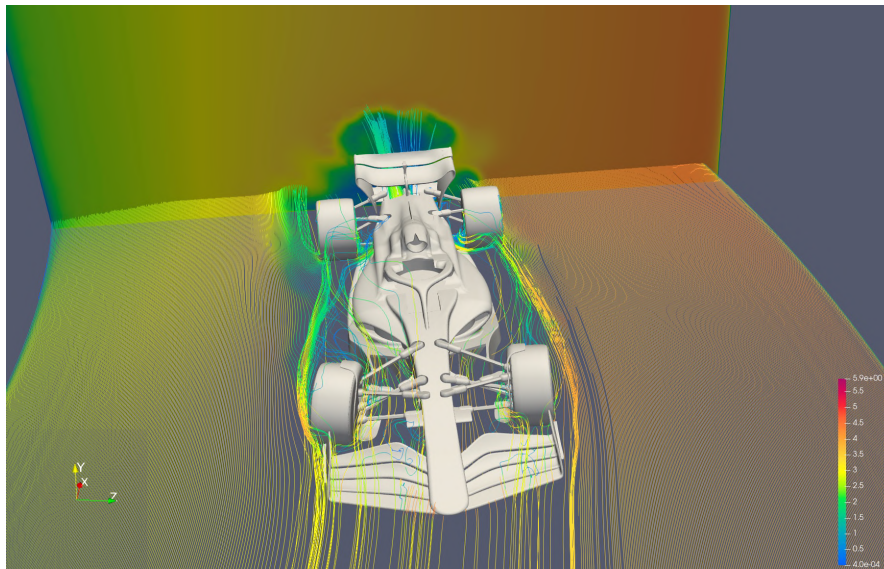


Figure 3.10: Velocity stream lines near the ground leading to the wake of the rear wheel.

3.3.4 Floor

An important component in generating downforce in the new regulations is the floor. Figure 3.13 presents the flow under the vehicle. There is a clear difference in the velocity magnitude between the flow on the outside and inside of the corner. This difference is approximately 1 m/s . Another interesting observation is the distinct difference in the wake of the front wheels. As discussed in the previous section due to different turning of the wheels the flow separation happens much easier for the wheel with a sharper turning angle. The wavy form of the wake on the outside wheel is due to the interaction of various

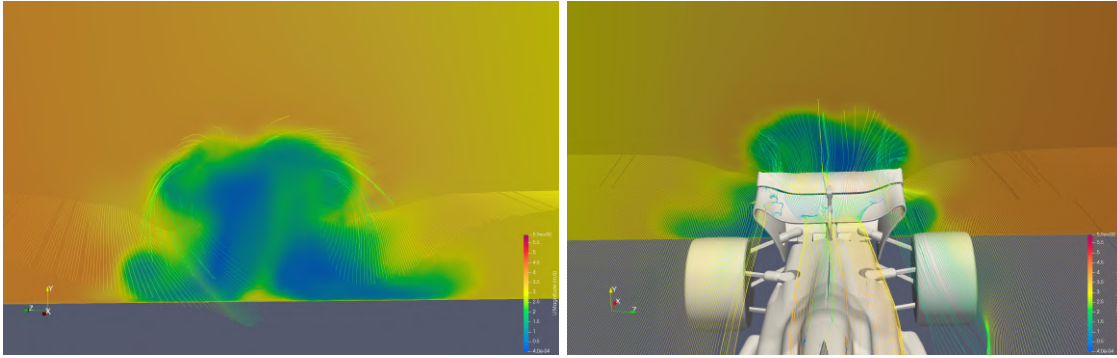


Figure 3.11: Rear wing velocity contour with streamline representation of the flow from the back (Left) and front (Right) of the car forming the outer part of the wake.

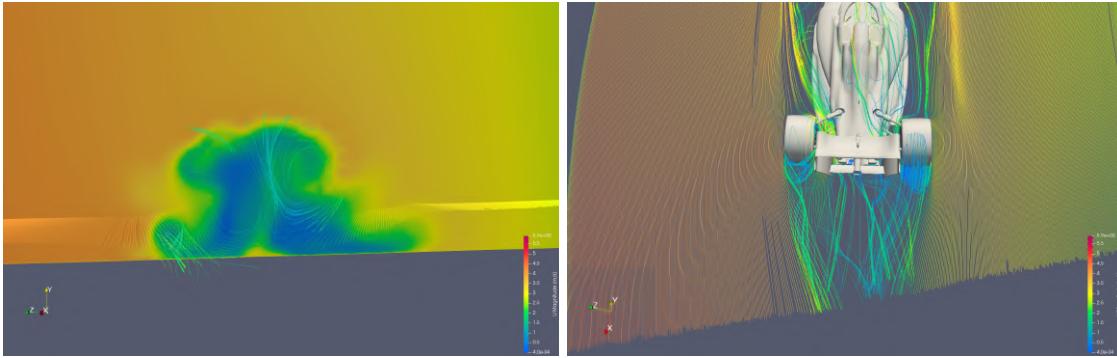


Figure 3.12: Rear wing velocity contour with streamline representation of the flow from the back of the car forming the inside of the wake.

aerodynamic components. Since this flow moves closer to the side pods and the edge of the floor, it is more likely to interact with the flow around the body. Additionally the flow out of the suspension components is another contributing factor which can be observed in figure 3.14.

3.4 Pressure coefficient

One of the most important aerodynamic parameters for F1 cars is the pressure coefficient. The latter can provide valuable insight on regions with adverse and favorable pressure gradient. Figure 3.15 presents the static pressure coefficient on body of the car. As expected, the pressure coefficient is quite high on the upper surface of the wings due to the higher pressure in these regions.

An interesting region is the front face of junctions of the halo with the body work. For a standard F1 car, the pressure coefficient in these regions is always positive since the velocity is minimum on these faces. However, in this case it can be observed that the left junction has a negative pressure coefficient. This is caused by the cylindrical housing

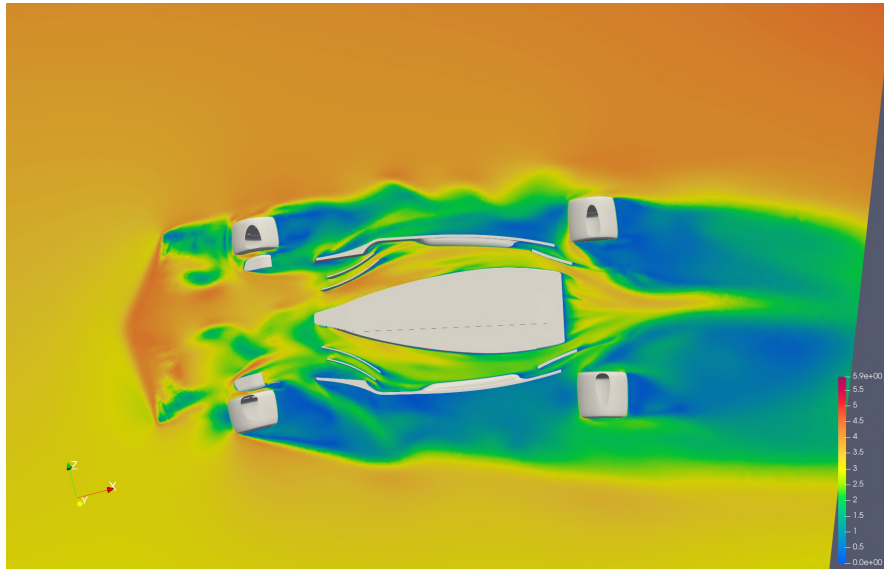


Figure 3.13: Velocity field under the car.

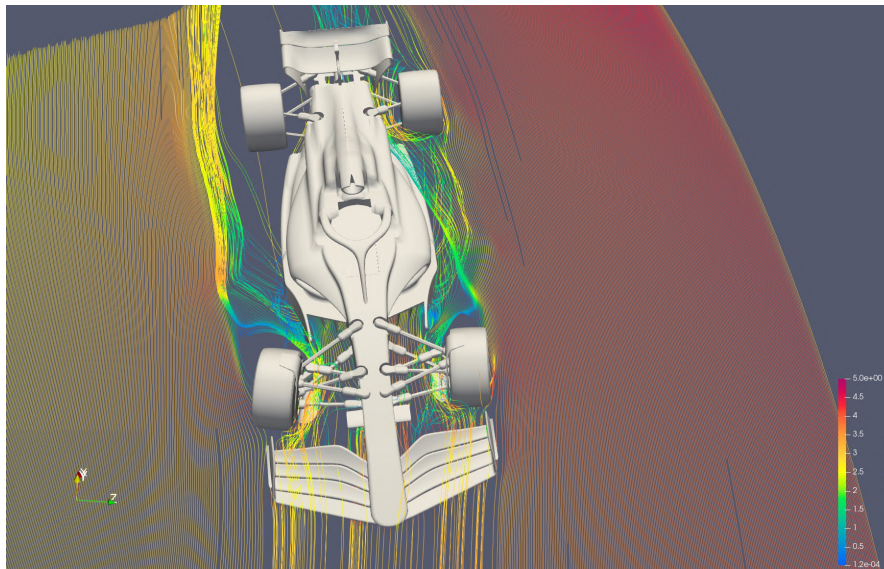


Figure 3.14: Streamlines around the wheels and suspensions.

created for fitting the arm and force measurement devices. In this part, the high velocity flow coming from the right side circulates inside the housing and guides the lower velocity air away from the halo junction, as a result a favorable pressure gradient is created leading to a negative pressure coefficient of -0.4 . This is visualised in figure 3.16.

Figure 3.17 shows the static pressure coefficient on the car floor. The pressure coefficient is approximately 1 at the entry of the floor tunnels. As the air accelerates through the floor this value gradually decreases. Around the diffusers this value is approximately -0.5 .

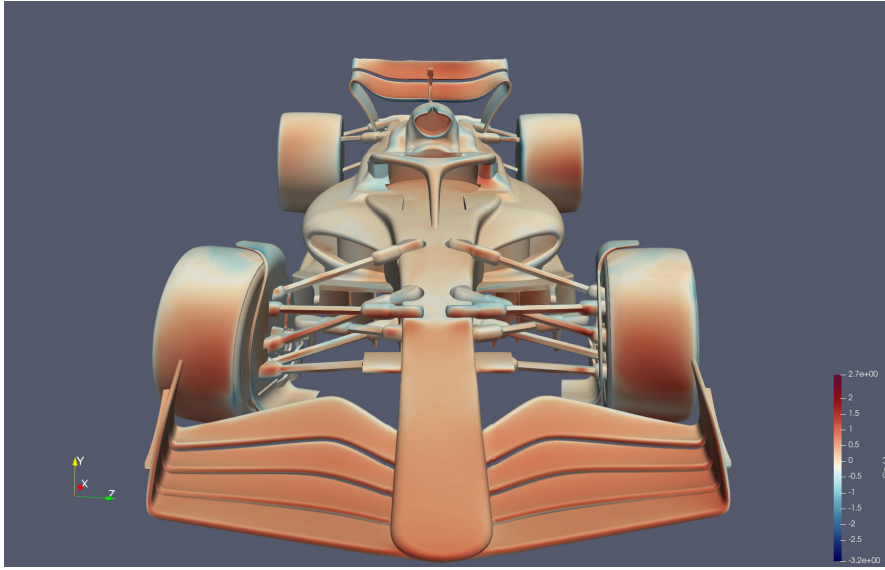


Figure 3.15: Pressure coefficient over the surface of the car.

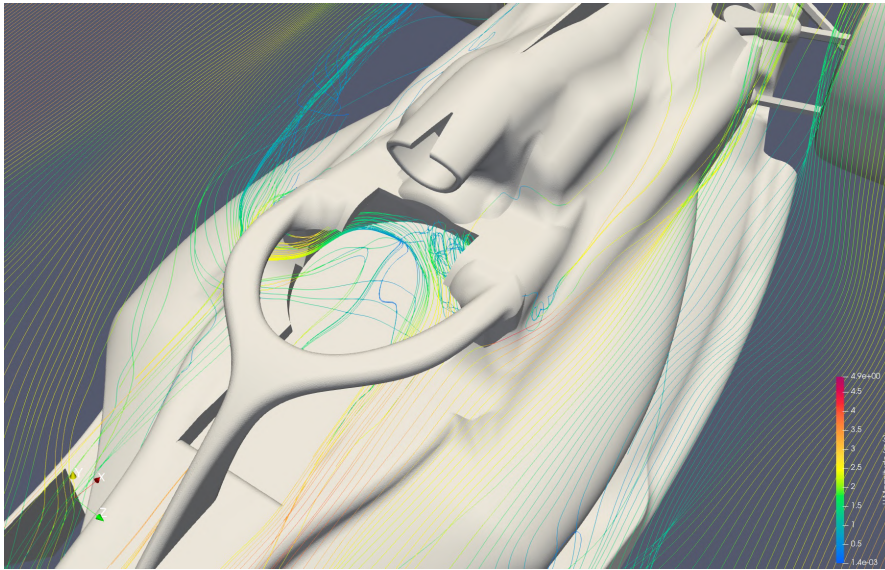


Figure 3.16: Visualisation of streamlines around the halo and the cylindrical housing.

3.5 PIV comparison

In order to compare the work presented in this report and validate the general flow trends, Particle Image Velocimetry (PIV) was carried out. Two planes were placed behind the rear wing perpendicular to the general flow direction. Figures 3.18 and 3.19 present the PIV domain. It should be noted that the experiments have been carried out over a short time period which increase the likelihood of measurement errors.

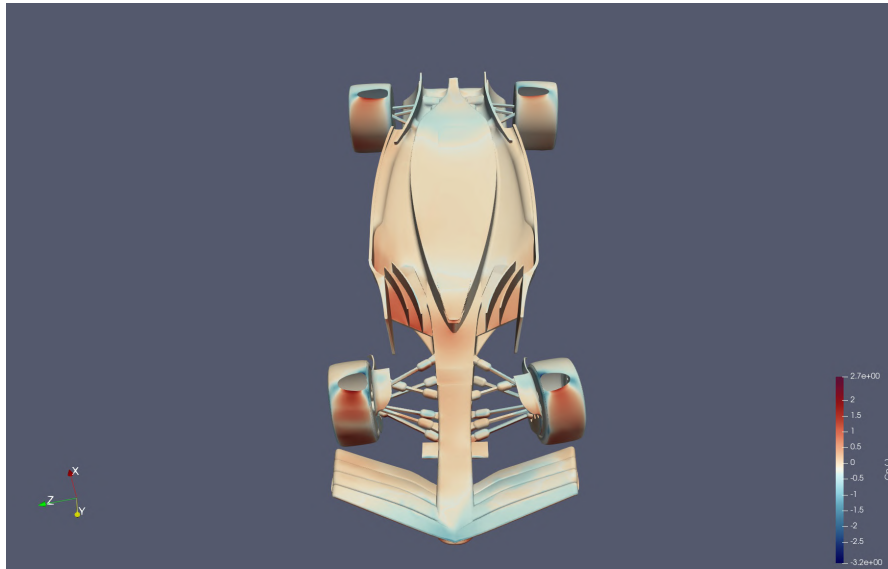


Figure 3.17: Pressure coefficient over the surface of the underbody.

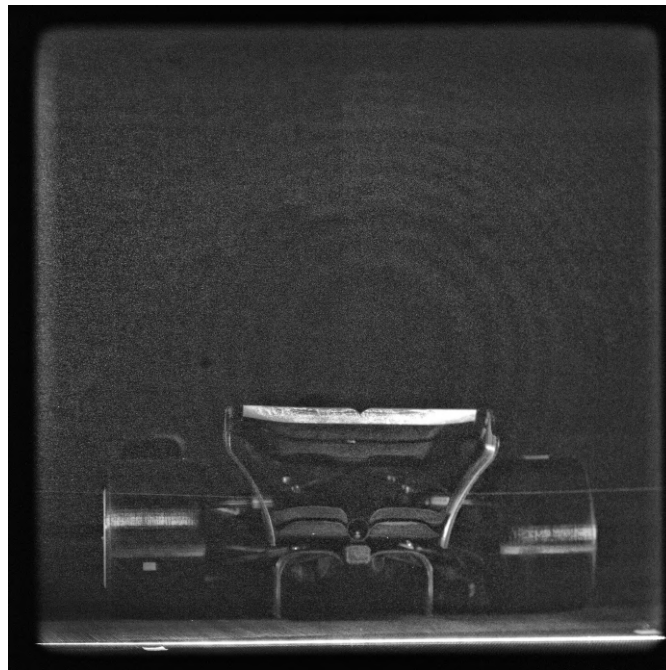


Figure 3.18: Picture of the car according to plane 1 configurations.

3.5.1 Plane 1: wake of the car

This plane was placed 10 *cm* behind the rear wing which was far enough from the car to capture the full wake of the car. Figure 3.20 shows the comparison between the PIV and simulation results. It is worth noting that the in-plane velocity magnitude is presented here. It can be observed that the maximum velocity according to the simulation is

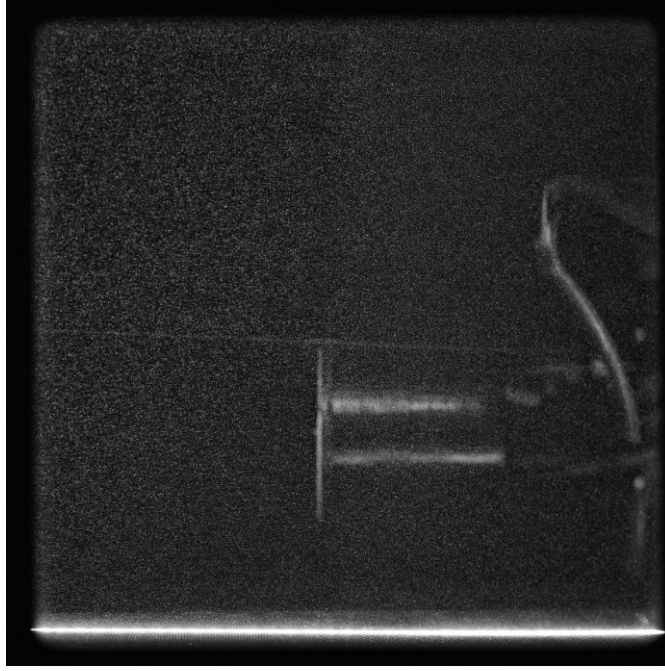


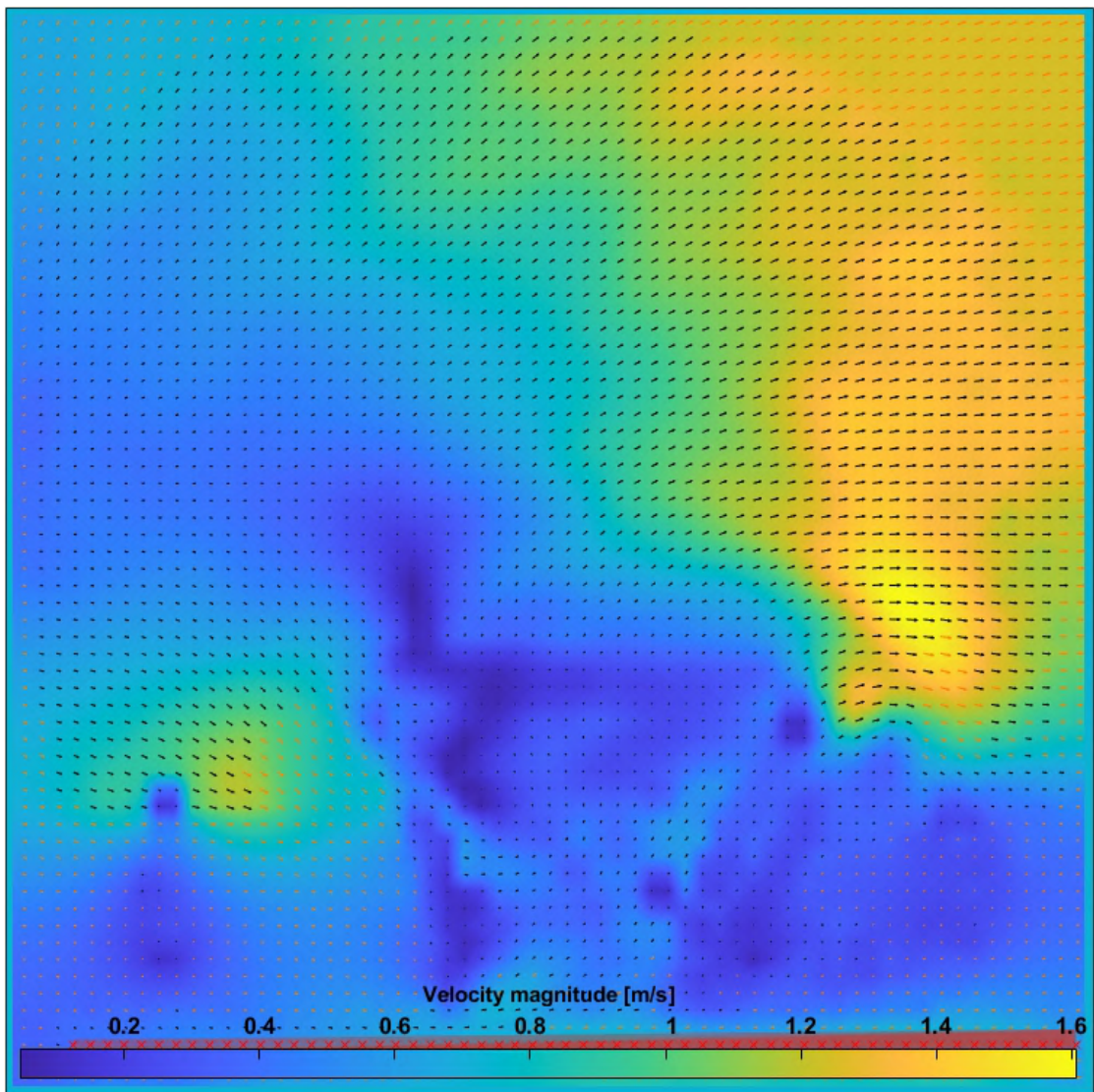
Figure 3.19: Picture of the car according to plane 2 configurations.

approximately 2 m/s while this value from PIV results is around 1.6 m/s which is a 25% error. Despite this difference, the flow is well replicated by the simulation. On the left side the vortices from the wake of the left wheel and the high velocity region can be seen on both results.

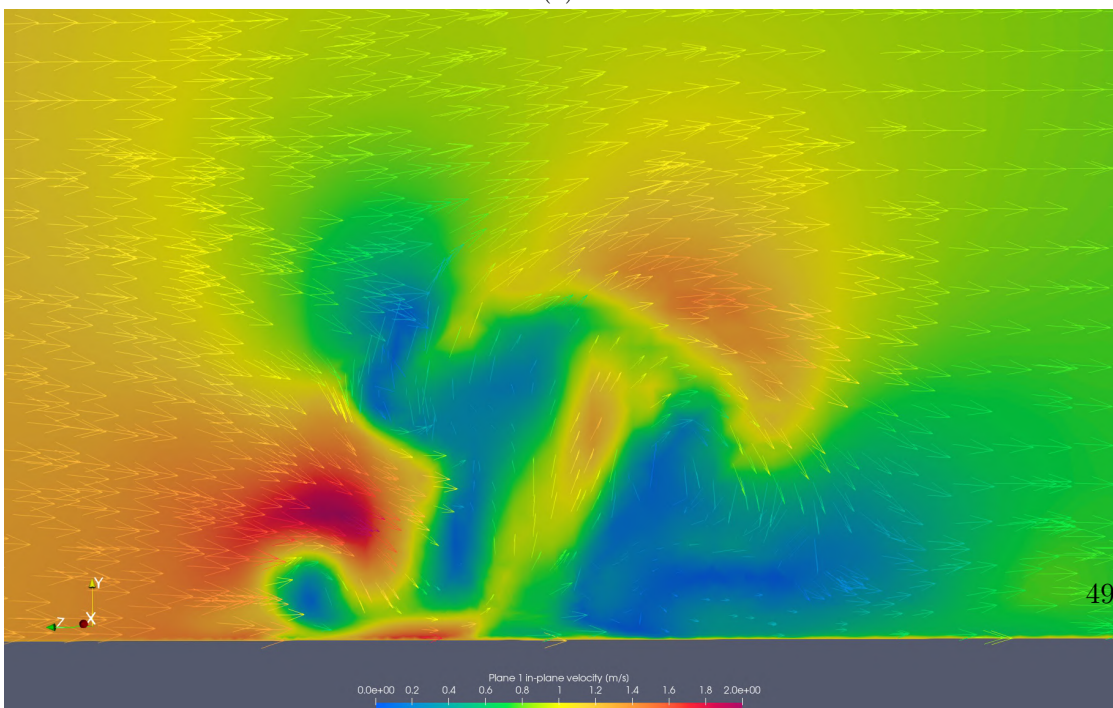
On the upper right side, the high velocity flow and its direction is also well predicted by the simulation. The velocity obtained by PIV is in the range of 1.2 m/s to 1.4 m/s while for the simulation this is in the range of 1.2 m/s to 1.5 m/s . The PIV velocity in the middle of the plane where the flow goes through the wing is around 0.6 m/s to 0.8 m/s while in CFD this is over a larger range from 0.4 m/s to 1.2 m/s . Another region that is partially well predicted is the wake of the left upper side of the wing. In this part the velocity is also better predicted.

3.5.2 Plane 2: wake of the car

This plane was placed 17 cm from the edge of the rear right wheel. Among the two PIV experiments, this experiment yielded the best data since the plane was closer to the rear wing and allowed better capture of the particle movements. The results are presented in figure 3.21. In this case the maximum velocity error is reduced to 11%. Similar to plane 1, the wake of the wheel and the upper edge of the wing is well predicted. It is also worth noting that due to limitations of the experiments the velocity and the wake in the lower regions near the ground are not well captured.

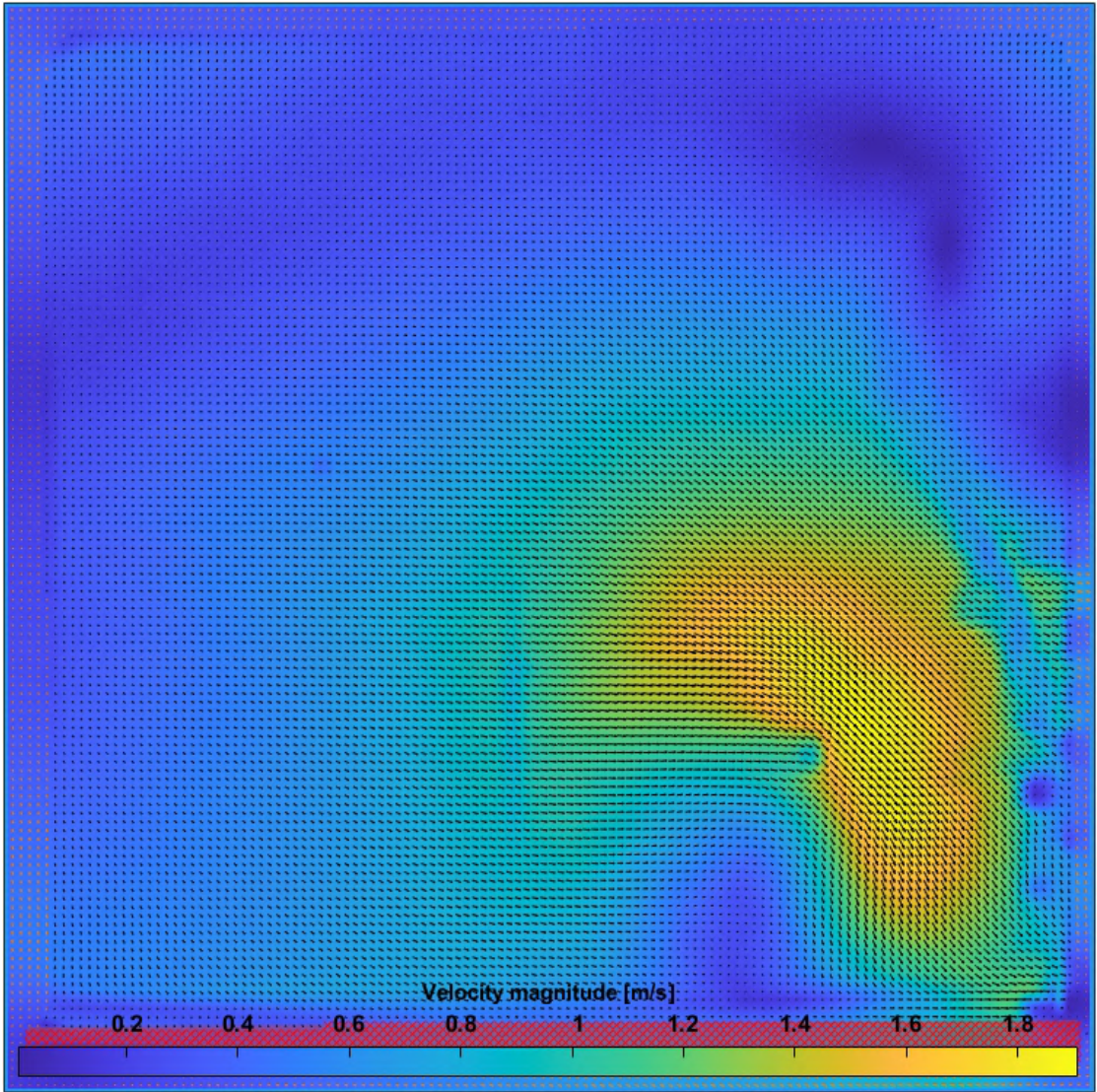


(a)

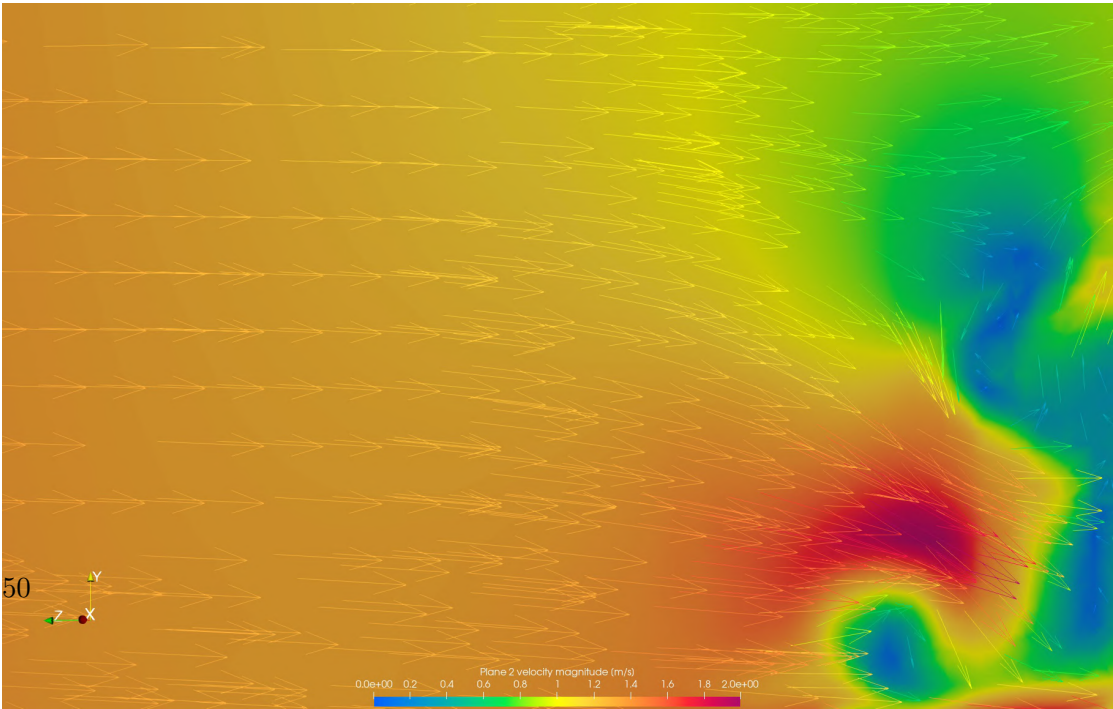


(b)

Figure 3.20: Result of the PIV experiment (a) and CFD simulations (b) at plane 1.



(a)



(b)

Figure 3.21: Result of the PIV experiment (a) and CFD simulations (b) at plane 2.

4 Conclusion

4.1 Summary

The goal of this project was to create a CFD platform for the existing test bench and validate it using PIV data. The first part of this study concentrated on creating CAD model adapted to CFD simulations. During this process necessary simplifications and modifications were made to improve the quality of the model for meshing.

The second part of the study focused on meshing and grid independence study. In this part the simulation parameters were carefully setup to achieve a stable simulation. It was observed that residuals for vertical velocity, turbulent kinetic energy, and pressure did not fall below the specified threshold. However, possible solutions such as adjusting the relaxation factors and allowing the simulation to run for an extended period of time proved to improve the residuals. The study of lift and drag coefficient over the body showed that the drag coefficient has stable behaviour and becomes independent of the number of grid elements with the studied meshes. The lift coefficient showed oscillatory behaviour and was not stable enough. This provided valuable insight into the possible limitations of the computational domain.

The results of the simulation showed that due to differences between the model and F1 cars certain distinct behaviours can be observed. The velocity field within the diffuser is one of such differences. Since the current model required special places for the suspensions, this led to wavy surfaces at the diffuser which resulted in flow blockage. Another interesting difference was observed at the halo junctions with the body. In this case it was observed that the housing created for measurement devices caused the incoming high velocity flow to be pushed towards the lower velocity junction and as a result create a favorable pressure gradient.

The comparison of simulation results and PIV experiments showed that the simulation results have a good agreement with PIV. The general flow trends were well represented by the simulation results. Despite the latter the error in velocity magnitude for plane 1 was shown to be around 25% which is large. The latter could be due to measurement errors since the experiments were carried out over a short period of time. On the second plane

this error was reduced which proved that better measurement data could significantly improve the error margin.

4.2 Future work

The goal of this project was to create a CFD platform to simulate the wind tunnel. This platform can now be used to test new ideas and also improve the wind tunnel and the car model. However, there are still certain areas where the simulation can improve.

Despite adapting the initial CAD model for CFD simulations, the overall quality of the model could be improved. The car itself can use certain improvements such as a new front wing that is compliant or at least close to FIA standards. Additionally, the floor could be further improved by removing the suspension housings and readjusting the suspension positions. These changes can significantly improve the quality of the model and result in higher quality mesh.

As it was discussed, the limitation of the computational domain can lead to instability of the flow and result in oscillatory behaviour of the lift coefficient. As a result, certain modifications on the wind tunnel such as increasing the upstream and downstream lengths could be beneficial to both numerical and experimental accuracy. Due to the unsteady nature of the flow, transient simulations could be an interesting alternative to improve the accuracy and stability of the simulation.

Bibliography

- [1] J. Keogh, T. Barber, S. Diasinos, and G. Doig, “Techniques for aerodynamic analysis of cornering vehicles,” *SAE Technical Paper Series*, 2015.
- [2] L. Konozy, *A new hypothesis on the anisotropic Reynolds stress tensor for turbulent flows volume I: Theoretical background and development of an anisotropic hybrid K-Omega Shear-stress transport/stochastic turbulence model*. Springer International Publishing, 2019.
- [3] “The menter shear stress transport turbulence model.” [Online]. Available: <https://turbmodels.larc.nasa.gov/sst.html>
- [4] “Crash introduction to the fvm and numerical playground, theory by joel guerrero.” [Online]. Available: https://wiki.openfoam.com/Theory_by_Joel_Guerrero
- [5] “What is a good mesh?” [Online]. Available: http://www.wolfdynamics.com/wiki/meshing_preliminaries_and_quality_assessment.pdf
- [6] S. Diasinos, G. Doig, J. Keogh, and T. Barber, “Detached eddy simulation of the cornering aerodynamics of the ahmed reference model,” 2014, FISITA World Automotive Congress (35th : 2014) ; Conference date: 02-06-2014 Through 06-06-2014.
- [7] E. Buri, “Etude aérodynamique de l’effet de sol d’une formule 1 en virage,” June 2022.
- [8] “Camber angle,” Aug 2023. [Online]. Available: https://en.wikipedia.org/wiki/Camber_angle
- [9] “Toe (automotive),” Dec 2022. [Online]. Available: [https://en.wikipedia.org/wiki/Toe_\(automotive\)](https://en.wikipedia.org/wiki/Toe_(automotive))
- [10] J. Vogel, “Tech explained: Ackermann steering geometry,” Apr 2021. [Online]. Available: <https://racecar-engineering.telegraph.co.uk/articles/tech-explained-ackermann-steering-geometry/#:~:text=The%20Ackermann%20level%20in%20a,or%20100%25%20Anti%2DAckermann.>

Bibliography

- [11] F. Moukalled, L. Mangani, and M. Darwish, *The finite volume method in Computational Fluid Dynamics: An advanced introduction with openfoam and MATLAB*. Springer, 2016.
- [12] “Helvetios specs.” [Online]. Available: <https://www.epfl.ch/research/facilities/scitas/hardware/helvetios/>
- [13] C. Greenshields, “Openfoam v10 user guide,” Jul 2023. [Online]. Available: <https://doc.cfd.direct/openfoam/user-guide-v10/standard-solvers>
- [14] “Simplefoam.” [Online]. Available: <https://openfoamwiki.net/index.php/SimpleFoam>
- [15] “Standard boundary conditions.” [Online]. Available: <https://www.openfoam.com/documentation/user-guide/a-reference/a.4-standard-boundary-conditions>
- [16] “Why menter’s sst model low-re issue has not been seriously investigated?” [Online]. Available: <https://www.cfd-online.com/Forums/openfoam/97520-why-menters-sst-model-low-re-issue-has-not-been-seriously-investigated.html>
- [17] “omegawallfunction.” [Online]. Available: <https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-wall-turbulence-omegaWallFunction.html>
- [18] “Attempt to cast type zerogradient to type nutwallfunction.” [Online]. Available: <https://develop.openfoam.com/Development/openfoam/-/issues/1966>
- [19] “Error with k-omega sst and simplefoam.” [Online]. Available: <https://www.cfd-online.com/Forums/openfoam-solving/224476-error-k-omega-sst-simplefoam.html>
- [20] “Gradient schemes.” [Online]. Available: <https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-gradient.html>
- [21] “Openfoam guide/limiters.” [Online]. Available: https://openfoamwiki.net/index.php/OpenFOAM_guide/Limiters
- [22] “Divergence schemes.” [Online]. Available: <https://www.openfoam.com/documentation/guides/v2112/doc/guide-schemes-divergence.html>
- [23] C. Greenshields, “Openfoam v11 user guide,” Jul 2023. [Online]. Available: <https://doc.cfd.direct/openfoam/user-guide-v11/fvschemes>
- [24] “26.2.1 first-order accuracy vs. second-order accuracy.” [Online]. Available: <https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node779.htm>
- [25] “Laplacian schemes.” [Online]. Available: <https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-laplacian.html>

- [26] “sngrad schemes.” [Online]. Available: <https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-sngrad.html>
- [27] C. Greenshields, “Openfoam v11 user guide,” Jul 2023. [Online]. Available: <https://doc.cfd.direct/openfoam/user-guide-v11/fvsolution>
- [28] M. Bhotvawala, “What’s smooth about the “smoothers” in cfd?” Nov 2021. [Online]. Available: <https://mustafabhotvawala.com/whats-smooth-about-the-smoothers-in-cfd/>
- [29] “Ansys mosaic meshing.” [Online]. Available: <https://www.ansys.com/products/fluids/ansys-fluent/mosaic-meshing>
- [30] “What are the methods for boundary layer mesh generation.” [Online]. Available: <https://courses.ansys.com/index.php/courses/add-boundary-layers/lessons/what-are-the-methods-for-boundary-layer-mesh-generation-lesson-2/>
- [31] “What is mesh stairstep?” [Online]. Available: <https://forum.ansys.com/forums/topic/inflation-created-stairstep-mesh-at-some-location/>
- [32] “Y+ boundary layer thickness,” May 2023. [Online]. Available: <https://resources.system-analysis.cadence.com/blog/msa2023-y-boundary-layer-thickness>
- [33] D. C. Wilcox, *Turbulence modeling for CFD*. DCW Industries, 2010.
- [34] J. M. Keogh, T. Barber, S. Diasinos, and G. Doig, “A new type of wind tunnel for the evaluation of curved motion,” *54th AIAA Aerospace Sciences Meeting*, 2016.
- [35] Gestione, “Cfd computational domain,” Apr 2022. [Online]. Available: <https://www.idealsimulations.com/resources/cfd-computational-domain/#>
- [36] J. Keogh, T. Barber, S. Diasinos, and G. Doig, “The aerodynamic effects on a cornering ahmed body,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 154, p. 34–46, 2016.

A Appendix

A.1 $k - \omega$ SST variables [2] [3]

- Term for production of turbulent energy.

$$\mathbf{P} = \tau_{ij} \frac{\partial u_i}{\partial x_j} \quad (\text{A.1})$$

- Reynolds stress tensor

$$\tau_{ij} = \mu_t \left(2S_{ij} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (\text{A.2})$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (\text{A.3})$$

- Menter's dynamic eddy viscosity coefficient

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, \Omega F_2)} \quad (\text{A.4})$$

- Ω is the vorticity magnitude

$$\Omega = \sqrt{2W_{ij}W_{ij}} \quad (\text{A.5})$$

$$W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (\text{A.6})$$

- Menter's first blending function

$$F_1 = \tanh(\arg_1^4) \quad (\text{A.7})$$

$$\arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d^2 \omega} \right), \frac{4\rho \sigma_\omega 2k}{CD_{k\omega} d^2} \right] \quad (\text{A.8})$$

$$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right) \quad (\text{A.9})$$

- Menter's second blending function

$$F_2 = \tanh(\arg_2^2) \quad (\text{A.10})$$

$$\arg_2 = \max \left(2 \frac{\sqrt{k}}{\beta^* \omega d}, \frac{500\nu}{d(2)\omega} \right) \quad (\text{A.11})$$

- d is the normal distance from the nearest wall
- The constants β , σ_k , and σ_ω are calculated using the following formula where ϕ is replace with the variable to be found:

$$\phi = F_1\phi_1 + (1 - F_1)\phi_2 \quad (\text{A.12})$$

$$\sigma_{k1} = 0.85, \sigma_{\omega 1} = 0.65, \beta_1 = 0.075 \quad (\text{A.13})$$

$$\sigma_{k2} = 1, \sigma_{\omega 2} = 0.856, \beta_2 = 0.0828 \quad (\text{A.14})$$

- $\beta^* = 0.09$ and $a_1 = 0.31$

A.2 decomposeParDict settings

Listing A.1: Controldict

```

1  /*-----*- C++ -*-----*/
2  / ===== /
3  / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4  / \ \ / O p e r a t i o n / Version: v2212 /
5  / \ \ / A n d / Website: www.openfoam.com /
6  / \ \ / M a n i p u l a t i o n / /
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        decomposeParDict;
14 }
15 // * * * * *
16
17 numberOfSubdomains 24;
18
19 method            simple;
20
21 simpleCoeffs
22 {
23     n              (4 3 2);
24     delta          0.001;
25 }
26
27
28 // * * * * *

```

A.3 ControlDict settings

Listing A.2: Controldict

```

1  /*-----*- C++ -*-----*/
2  /
3  /           Generated by the CfdOF workbench for FreeCAD /
4  /           https://gitlab.com/opensimproject/CfdOF /
5  /
6  \*-----*/
7  FoamFile
8  {
9     version      2.0;
10     format        ascii;
11     class         dictionary;
12     object        controlDict;
13 }
14 // * * * * *
15
16 application      simpleFoam;
17
18 startFrom        startTime;
19
20 startTime        0;

```

Appendix A. Appendix

```
21
22  stopAt          endTime;
23
24  deltaT          1;
25
26  endTime         7000;
27
28  writeControl     timeStep;
29
30  writeInterval    500;
31
32  purgeWrite       0;
33
34  writeFormat      ascii;
35
36  writePrecision   8;
37
38  runTimeModifiable true;
39
40
41
42
43  libs (
44      "libOpenFOAM.so"
45
46      "libswakFunctionObjects.so"
47      "libgroovyBC.so"
48  );
49
50
51
52  functions
53  {
54      force_coefficient
55      {
56          type          forceCoeffs;
57          libs          ("libforces.so");
58          patches        (front-left-wheel wing-fillets left-deflector left-
59                          suspension front-right-wheel right-deflector front-right-suspension
60                          front-wing rear-left-wheel rear-left-suspension rear-right-wheel rear-
61                          right-suspension rear-wing chassis left-suspension:1 front-right-
62                          suspension:1 winglets enclosure-enclosure:1);
63
64          writeControl    timeStep;
65          writeInterval   1;
66
67          p              p;
68          U              U;
69          rho            rhoInf;
70          rhoInf         1.225;
71
72          CofR           (0.8555 0.0681 0.6372);
73          liftDir         (0 1 0);
74          dragDir         (0.9871 0 0.2079);
75
76          pitchAxis      (0 1 0);
77
78          magUInf         3.3;
79
80          lRef           0.6014;
```

```
77         Aref          0.0520678;
78
79     }
80
81     yplus
82     {
83         type            yPlus;
84         libs             ("libfieldFunctionObjects.so");
85         writeControl     timeStep;
86         writeInterval    1000;
87
88     }
89 }
90
91
92
93
94 // ***** //
```

A.4 Boundary conditions

A.4.1 U

Listing A.3: U

```

1  /*-----*- C++ -*-----*\
2  /
3  /      Generated by the CfdOF workbench for FreeCAD
4  /      https://gitlab.com/opensimproject/CfdOF
5  /
6  \*-----*/
7  FoamFile
8  {
9      version      2.0;
10     format        ascii;
11     class          volVectorField;
12     object          U;
13 }
14 // * * * * *
15
16 dimensions        [0 1 -1 0 0 0 0];
17
18 internalField      uniform (0.0 0.0 0.0);
19
20 boundaryField
21 {
22
23     inlet
24     {
25         type          groovyBC;
26         variables
27         (
28             "zp2=pos().z;"
29
30
31             "w=0.88;"
32
33             "U_tx=w*zp2+2.64;"
34             "U_tz=0;"
35             "U_ax=0;"
36         );
37         valueExpression "vector(U_tx, U_ax, U_tz)";
38         value uniform (0 0 0);
39     }
40
41     ground
42     {
43
44         type          rotatingWallVelocity;
45         origin          (0 0 -3);
46         axis            (0 1 0);
47         omega           0.88;
48         value           uniform (0 0 0);
49     }
50
51     front-left-wheel
52     {
53         type          rotatingWallVelocity;

```

```

54         origin      (0.5094676 0.0646096 0.5533705);
55         axis        (-0.00267157 -0.061 0.99791415);
56         omega       51.59;
57         value       uniform (0 0 0);
58
59     }
60
61     front-right-wheel
62     {
63
64         type         rotatingWallVelocity;
65         origin      (0.5737 0.0636 0.8556);
66         axis        (0.095 0.061 0.9932);
67         omega       55.72;
68         value       uniform (0 0 0);
69     }
70
71     rear-left-wheel
72     {
73
74         type         rotatingWallVelocity;
75         origin      (1.0981 .0634 0.4304);
76         axis        (0.2331 -0.0261 0.9719);
77         omega       51.59;
78         value       uniform (0 0 0);
79     }
80
81     rear-right-wheel
82     {
83
84         type         rotatingWallVelocity;
85         origin      (1.1612 .0634 0.7272);
86         axis        (0.1823 0.0261 0.9827);
87         omega       55.72;
88         value       uniform (0 0 0);
89     }
90
91     outlet
92     {
93         type         zeroGradient;
94
95
96     }
97
98
99     tunnel-walls
100    {
101        type         noSlip;
102
103    }
104
105    radii
106    {
107        type         noSlip;
108
109    }
110
111    left-deflector
112    {
113        type         noSlip;

```

Appendix A. Appendix

```
114     }
115
116
117     left-suspension
118     {
119         type          noSlip;
120
121     }
122
123     right-deflector
124     {
125         type          noSlip;
126
127     }
128
129     front-right-suspension
130     {
131         type          noSlip;
132
133     }
134
135     front-wing
136     {
137         type          noSlip;
138
139     }
140
141     rear-left-suspension
142     {
143         type          noSlip;
144
145     }
146
147     rear-right-suspension
148     {
149         type          noSlip;
150
151     }
152
153     rear-wing
154     {
155         type          noSlip;
156
157     }
158
159     chassis
160     {
161         type          noSlip;
162
163     }
164
165     left-suspension:159
166     {
167         type          noSlip;
168
169     }
170
171     front-right-suspension:160
172     {
173         type          noSlip;
```



```

174
175     }
176
177     wing-fillets
178     {
179         type            noSlip;
180
181     }
182
183     winglets
184     {
185         type            noSlip;
186
187     }
188
189     chassis-spaces
190     {
191         type            noSlip;
192
193     }
194
195     wing-spaces
196     {
197         type            noSlip;
198
199     }
200
201     wheel-holes
202     {
203         type            noSlip;
204
205     }
206
207 }
208
209
210 // *****

```

A.4.2 p

Listing A.4: p

```

1      /*-----*- C++
      -*-----*\
2      /
3      /          Generated by the CfdOF workbench for FreeCAD
4      /          https://gitlab.com/opensimproject/CfdOF
5      /
6      \*-----*/
7      FoamFile
8      {
9          version      2.0;
10         format       ascii;
11         class         volScalarField;
12         object        p;
13     }
14 // * * * * *
15

```

Appendix A. Appendix

```
16 dimensions      [0 2 -2 0 0 0 0];
17
18 internalField    uniform 0.0;
19
20 boundaryField
21 {
22
23     inlet
24     {
25         type      zeroGradient;
26     }
27
28     front-left-wheel
29     {
30         type      zeroGradient;
31     }
32
33     outlet
34     {
35         type      fixedValue;
36         value     uniform 0;
37     }
38
39     radii
40     {
41         type      zeroGradient;
42     }
43
44     tunnel-walls
45     {
46         type      zeroGradient;
47     }
48
49     ground
50     {
51         type      zeroGradient;
52     }
53
54     left-deflector
55     {
56         type      zeroGradient;
57     }
58
59     left-suspension
60     {
61         type      zeroGradient;
62     }
63
64     front-right-wheel
65     {
66         type      zeroGradient;
67     }
68
69     right-deflector
70     {
71         type      zeroGradient;
72     }
73
74     front-right-suspension
75     {
```

```

76         type          zeroGradient;
77     }
78
79     front-wing
80     {
81         type          zeroGradient;
82     }
83
84     rear-left-wheel
85     {
86         type          zeroGradient;
87     }
88
89     rear-left-suspension
90     {
91         type          zeroGradient;
92     }
93
94     rear-right-wheel
95     {
96         type          zeroGradient;
97     }
98
99     rear-right-suspension
100    {
101        type          zeroGradient;
102    }
103
104    rear-wing
105    {
106        type          zeroGradient;
107    }
108
109    chassis
110    {
111        type          zeroGradient;
112    }
113
114    left-suspension:159
115    {
116        type          zeroGradient;
117    }
118
119    front-right-suspension:160
120    {
121        type          zeroGradient;
122    }
123
124    wheel-holes
125    {
126        type          zeroGradient;
127    }
128
129    wing-spaces
130    {
131        type          zeroGradient;
132    }
133
134    chassis-spaces
135    {

```

Appendix A. Appendix

```
136         type          zeroGradient;
137     }
138
139     winglets
140     {
141         type          zeroGradient;
142     }
143
144     wing-fillets
145     {
146         type          zeroGradient;
147     }
148
149
150 }
151
152 // ***** //
```

A.4.3 k

Listing A.5: k

```
1  /*-----* C++ -*-----*\
2  /
3  /      Generated by the CfdOF workbench for FreeCAD      /
4  /      https://gitlab.com/opensimproject/CfdOF          /
5  /
6  \*-----*/
7  FoamFile
8  {
9      version      2.0;
10     format       ascii;
11     class        volScalarField;
12     object       k;
13 }
14 // * * * * * //
15
16 dimensions      [0 2 -2 0 0 0 0];
17
18 internalField    uniform 1e-10;
19
20 boundaryField
21 {
22
23     inlet
24     {
25         type      fixedValue;
26         value     $internalField;
27     }
28
29     ground
30     {
31         type      fixedValue;
32         value     $internalField;
33     }
34
35     outlet
36     {
```

```

37         type          fixedValue;
38         value          $internalField;
39
40     }
41
42     tunnel-walls
43     {
44         type          fixedValue;
45         value          $internalField;
46
47     }
48
49     radii
50     {
51         type          fixedValue;
52         value          $internalField;
53     }
54
55     front-left-wheel
56     {
57         type          fixedValue;
58         value          $internalField;
59     }
60
61     left-deflector
62     {
63         type          fixedValue;
64         value          $internalField;
65     }
66
67     left-suspension
68     {
69         type          fixedValue;
70         value          $internalField;
71     }
72
73     front-right-wheel
74     {
75         type          fixedValue;
76         value          $internalField;
77     }
78
79     right-deflector
80     {
81         type          fixedValue;
82         value          $internalField;
83     }
84
85     front-right-suspension
86     {
87         type          fixedValue;
88         value          $internalField;
89     }
90
91     front-wing
92     {
93         type          fixedValue;
94         value          $internalField;
95     }
96

```

Appendix A. Appendix

```
97     rear-left-wheel
98     {
99         type          fixedValue;
100        value          $internalField;
101    }
102
103     rear-left-suspension
104     {
105         type          fixedValue;
106        value          $internalField;
107    }
108
109     rear-right-wheel
110     {
111         type          fixedValue;
112        value          $internalField;
113    }
114
115     rear-right-suspension
116     {
117         type          fixedValue;
118        value          $internalField;
119    }
120
121     rear-wing
122     {
123         type          fixedValue;
124        value          $internalField;
125    }
126
127     chassis
128     {
129         type          fixedValue;
130        value          $internalField;
131    }
132
133     left-suspension:159
134     {
135         type          fixedValue;
136        value          $internalField;
137    }
138
139     front-right-suspension:160
140     {
141         type          fixedValue;
142        value          $internalField;
143    }
144
145     wing-fillets
146     {
147         type          fixedValue;
148        value          $internalField;
149    }
150
151     winglets
152     {
153         type          fixedValue;
154        value          $internalField;
155    }
156
```

```

157
158     chassis-spaces
159     {
160         type            fixedValue;
161         value            $internalField;
162     }
163
164     wing-spaces
165     {
166         type            fixedValue;
167         value            $internalField;
168     }
169
170     wheel-holes
171     {
172         type            fixedValue;
173         value            $internalField;
174     }
175 }
176
177
178 // *****

```

A.4.4 omega

Listing A.6: omega

```

1  /*-----*- C++ -*-----*\
2  / ===== /
3  / \ \ / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
4  / \ \ / O p e r a t i o n / V e r s i o n : v 2 2 1 2 /
5  / \ \ / A n d / W e b s i t e : w w w . o p e n f o a m . c o m /
6  / \ \ / M a n i p u l a t i o n /
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     object        omega;
14 }
15 // * * * * *
16
17
18 dimensions      [0 0 -1 0 0 0 0];
19
20 internalField    uniform 0.01;
21
22 boundaryField
23 {
24
25     inlet
26     {
27         type      fixedValue;
28         value      $internalField;
29     }
30 }
31

```

Appendix A. Appendix

```
32     ground
33     {
34         type      omegaWallFunction;
35         value     $internalField;
36
37     }
38
39     outlet
40     {
41         type      inletOutlet;
42         inletValue $internalField;
43         value     $internalField;
44
45     }
46
47     tunnel-walls
48     {
49         type      omegaWallFunction;
50         value     $internalField;
51
52     }
53
54     radii
55     {
56
57         type      omegaWallFunction;
58         value     $internalField;
59
60     }
61
62     front-left-wheel
63     {
64         type      omegaWallFunction;
65         value     $internalField;
66
67     }
68
69     left-deflector
70     {
71         type      omegaWallFunction;
72         value     $internalField;
73
74     }
75
76     left-suspension
77     {
78         type      omegaWallFunction;
79         value     $internalField;
80
81     }
82
83     front-right-wheel
84     {
85         type      omegaWallFunction;
86         value     $internalField;
87
88     }
89
90
91
```



```

92     }
93
94     right-deflector
95     {
96         type      omegaWallFunction;
97         value      $internalField;
98
99     }
100
101
102     front-right-suspension
103     {
104         type      omegaWallFunction;
105         value      $internalField;
106
107     }
108
109
110     front-wing
111     {
112         type      omegaWallFunction;
113         value      $internalField;
114
115     }
116
117
118     rear-left-wheel
119     {
120         type      omegaWallFunction;
121         value      $internalField;
122
123     }
124
125
126     rear-left-suspension
127     {
128         type      omegaWallFunction;
129         value      $internalField;
130
131     }
132
133
134     rear-right-wheel
135     {
136         type      omegaWallFunction;
137         value      $internalField;
138
139     }
140
141     rear-right-suspension
142     {
143         type      omegaWallFunction;
144         value      $internalField;
145
146     }
147
148     rear-wing
149     {
150         type      omegaWallFunction;
151         value      $internalField;

```

Appendix A. Appendix

```
152     }
153
154
155     chassis
156     {
157         type      omegaWallFunction;
158         value      $internalField;
159     }
160
161
162     left-suspension:159
163     {
164         type      omegaWallFunction;
165         value      $internalField;
166     }
167
168
169     front-right-suspension:160
170     {
171         type      omegaWallFunction;
172         value      $internalField;
173     }
174
175
176     wing-fillets
177     {
178         type      omegaWallFunction;
179         value      $internalField;
180     }
181
182
183     winglets
184     {
185         type      omegaWallFunction;
186         value      $internalField;
187     }
188
189
190     chassis-spaces
191     {
192         type      omegaWallFunction;
193         value      $internalField;
194     }
195
196
197     wing-spaces
198     {
199         type      omegaWallFunction;
200         value      $internalField;
201     }
202
203
204     wheel-holes
205     {
206         type      omegaWallFunction;
207         value      $internalField;
208     }
209
210
211
```

```

212 }
213
214
215 // *****

```

A.4.5 nut OpenFOAM v2112

Listing A.7: nut OpenFOAM v2112

```

1      /*----- C++
      -*-----*/
2      /
3      /          Generated by the CfdOF workbench for FreeCAD          /
4      /          https://gitlab.com/opensimproject/CfdOF              /
5      /
6      \*-----*/
7      FoamFile
8      {
9          version      2.0;
10         format        ascii;
11         class          volScalarField;
12         object          nut;
13     }
14     // * * * * *
15
16     dimensions          [0 2 -1 0 0 0 0];
17
18     internalField        uniform 0;
19
20     boundaryField
21     {
22
23         inlet
24         {
25             type          calculated;
26             value          $internalField;
27         }
28
29         ground
30         {
31             type          nutLowReWallFunction;
32             value          uniform 0;
33         }
34
35         outlet
36         {
37             type          calculated;
38             value          $internalField;
39         }
40
41         tunnel-walls
42         {
43             type          nutLowReWallFunction;
44             value          uniform 0;
45         }
46     }
47
48     radii

```

Appendix A. Appendix

```
49     {
50         type      nutLowReWallFunction;
51         value      uniform 0;
52     }
53
54
55     front-left-wheel
56     {
57         type      nutLowReWallFunction;
58         value      uniform 0;
59     }
60
61
62     left-deflector
63     {
64         type      nutLowReWallFunction;
65         value      uniform 0;
66     }
67
68
69     left-suspension
70     {
71         type      nutLowReWallFunction;
72         value      uniform 0;
73     }
74
75     front-right-wheel
76     {
77         type      nutLowReWallFunction;
78         value      uniform 0;
79     }
80
81     right-deflector
82     {
83         type      nutLowReWallFunction;
84         value      uniform 0;
85     }
86
87
88     front-right-suspension
89     {
90         type      nutLowReWallFunction;
91         value      uniform 0;
92     }
93
94
95     front-wing
96     {
97         type      nutLowReWallFunction;
98         value      uniform 0;
99     }
100
101
102     rear-left-wheel
103     {
104         type      nutLowReWallFunction;
105         value      uniform 0;
106     }
107
108
```

```

109     rear-left-suspension
110     {
111         type          nutLowReWallFunction;
112         value          uniform 0;
113     }
114
115
116     rear-right-wheel
117     {
118         type          nutLowReWallFunction;
119         value          uniform 0;
120     }
121
122
123     rear-right-suspension
124     {
125         type          nutLowReWallFunction;
126         value          uniform 0;
127     }
128
129
130     rear-wing
131     {
132         type          nutLowReWallFunction;
133         value          uniform 0;
134     }
135
136
137     chassis
138     {
139         type          nutLowReWallFunction;
140         value          uniform 0;
141     }
142
143
144     left-suspension:159
145     {
146         type          nutLowReWallFunction;
147         value          uniform 0;
148     }
149
150
151     front-right-suspension:160
152     {
153         type          nutLowReWallFunction;
154         value          uniform 0;
155     }
156
157
158     wing-fillets
159     {
160         type          nutLowReWallFunction;
161         value          uniform 0;
162     }
163
164
165     winglets
166     {
167         type          nutLowReWallFunction;
168         value          uniform 0;

```

Appendix A. Appendix

```
169     }
170
171
172     chassis-spaces
173     {
174         type      nutLowReWallFunction;
175         value      uniform 0;
176     }
177
178
179     wing-spaces
180     {
181         type      nutLowReWallFunction;
182         value      uniform 0;
183     }
184
185
186     wheel-holes
187     {
188         type      nutLowReWallFunction;
189         value      uniform 0;
190     }
191 }
192
193
194 // ***** //
```

A.4.6 nut OpenFOAM v2212

Listing A.8: nut OpenFOAM v2212

```
1  /*----- C++
   - *-----\
2  /
3  /      Generated by the CfdOF workbench for FreeCAD
4  /      https://gitlab.com/opensimproject/CfdOF
5  /
6  \ *-----*/
7  FoamFile
8  {
9      version      2.0;
10     format        ascii;
11     class         volScalarField;
12     object        nut;
13 }
14 // * * * * *
15
16 dimensions      [0 2 -1 0 0 0 0];
17
18 internalField    uniform 0;
19
20 boundaryField
21 {
22
23     inlet
24     {
25         type      calculated;
26         value      $internalField;
```

```

27     }
28
29     ground
30     {
31         type          calculated;
32         value          uniform 0;
33     }
34
35     outlet
36     {
37         type          calculated;
38         value          $internalField;
39     }
40
41     tunnel-walls
42     {
43         type          calculated;
44         value          uniform 0;
45     }
46
47
48     radii
49     {
50         type          calculated;
51         value          uniform 0;
52     }
53
54
55     front-left-wheel
56     {
57         type          calculated;
58         value          uniform 0;
59     }
60
61
62     left-deflector
63     {
64         type          calculated;
65         value          uniform 0;
66     }
67
68
69     left-suspension
70     {
71         type          calculated;
72         value          uniform 0;
73     }
74
75     front-right-wheel
76     {
77         type          calculated;
78         value          uniform 0;
79     }
80
81     right-deflector
82     {
83         type          calculated;
84         value          uniform 0;
85     }
86

```

Appendix A. Appendix

```
87
88     front-right-suspension
89     {
90         type          calculated;
91         value          uniform 0;
92     }
93
94     front-wing
95     {
96         type          calculated;
97         value          uniform 0;
98     }
99
100
101     rear-left-wheel
102     {
103         type          calculated;
104         value          uniform 0;
105     }
106
107
108     rear-left-suspension
109     {
110         type          calculated;
111         value          uniform 0;
112     }
113
114
115     rear-right-wheel
116     {
117         type          calculated;
118         value          uniform 0;
119     }
120
121
122     rear-right-suspension
123     {
124         type          calculated;
125         value          uniform 0;
126     }
127
128
129     rear-wing
130     {
131         type          calculated;
132         value          uniform 0;
133     }
134
135
136     chassis
137     {
138         type          calculated;
139         value          uniform 0;
140     }
141
142
143     left-suspension:1
144     {
145         type          calculated;
```



```

147         value      uniform 0;
148
149     }
150
151     front-right-suspension:1
152     {
153         type      calculated;
154         value      uniform 0;
155     }
156
157
158     wing-fillets
159     {
160         type      calculated;
161         value      uniform 0;
162     }
163
164
165     winglets
166     {
167         type      calculated;
168         value      uniform 0;
169     }
170
171
172     chassis-spaces
173     {
174         type      calculated;
175         value      uniform 0;
176     }
177
178
179     wing-spaces
180     {
181         type      calculated;
182         value      uniform 0;
183     }
184
185
186     wheel-holes
187     {
188         type      calculated;
189         value      uniform 0;
190     }
191 }
192
193
194 // ***** //
```

A.5 fvSchemes

Listing A.9: fvScheme

```

1      /*----- C++
      -*-----*/
2      / ===== /
3      / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4      / \ \ / O p e r a t i o n / Version: v2012 /
5      / \ \ / A n d / Website: www.openfoam.com /
6      / \ \ / M a n i p u l a t i o n / /
7      /*-----*/
8      FoamFile
9      {
10         version      2.0;
11         format        ascii;
12         class          dictionary;
13         object          fvSchemes;
14     }
15     // * * * * *
16
17     ddtSchemes
18     {
19         default          steadyState;
20     }
21
22     gradSchemes
23     {
24         default          Gauss linear;
25
26         grad(omega)      faceLimited Gauss linear 1;
27         grad(k)          faceLimited Gauss linear 1;
28     }
29
30
31     divSchemes
32     {
33         default          none;
34         div(phi,U)      Gauss linearUpwind grad(U);
35         div(phi,k)      Gauss linearUpwind grad(k);
36         div(phi,omega)  Gauss linearUpwind grad(omega);
37         div((nuEff*dev2(T(grad(U)))) Gauss linear;
38     }
39
40     laplacianSchemes
41     {
42         default          Gauss linear corrected;
43     }
44
45     interpolationSchemes
46     {
47         default          linear;
48     }
49
50     snGradSchemes
51     {
52         default          corrected;
53     }
54
55     wallDist

```

```

56 {
57     method meshWave;
58 }
59
60
61 // *****

```

A.6 fvSolutions

Listing A.10: fvSolution

```

1      /*----- C++
      -*-----*/
2      / ===== /
3      / \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
4      / \ \ / O p e r a t i o n / Version: v2012 /
5      / \ \ / A n d / Website: www.openfoam.com /
6      / \ \ / M a n i p u l a t i o n / /
7      \*-----*/
8      FoamFile
9      {
10         version      2.0;
11         format        ascii;
12         class          dictionary;
13         object          fvSolution;
14     }
15     // * * * * *
16
17     solvers
18     {
19         p
20         {
21             solver      GAMG;
22             smoother     GaussSeidel;
23             tolerance    1e-6;
24             relTol       0.01;
25         }
26
27         Phi
28         {
29             $p;
30         }
31
32         U
33         {
34             solver      smoothSolver;
35             smoother     GaussSeidel;
36             tolerance    1e-10;
37             relTol       0.1;
38             nSweeps      1;
39         }
40
41         k
42         {
43             solver      smoothSolver;
44             smoother     GaussSeidel;
45             tolerance    1e-20;
46             relTol       0.1;

```

Appendix A. Appendix

```
47         nSweeps          1;
48     }
49
50     omega
51     {
52         solver             smoothSolver;
53         smoother           GaussSeidel;
54         tolerance          1e-20;
55         relTol             0.1;
56         nSweeps            1;
57     }
58 }
59
60 SIMPLE
61 {
62     nNonOrthogonalCorrectors 2;
63     consistent yes;
64
65     residualControl
66     {
67         p                1e-3;
68         U                1e-3;
69         omega            1e-3;
70         k                1e-3;
71     }
72 }
73
74
75 potentialFlow
76 {
77     nNonOrthogonalCorrectors 10;
78 }
79
80 relaxationFactors
81 {
82     equations
83     {
84         U                0.7;    //after iteration 4500 change to 0.3
85         k                0.7;
86         omega            0.7;
87     }
88
89     fields
90     {
91         p                0.3; //between iterations 2000-3000 change to
92                               0.5 and at 4500 change to 0.7
93     }
94
95 }
96
97
98
99
100 cache
101 {
102     grad(U);
103 }
104
105 // *****
```

A.7 Y+ value

Table A.1: Y+ values for mesh 2.

Part	Min. Y+	Max. Y+	Avg. Y+
Rear Wing	0.0003	16.71	3.25
Front wing	0.006	14.02	3.01
Winglets	0.007	8.5	1.78
Chassis	$6 \cdot 10^{-5}$	15.47	2.93
Ground	0.093	18.15	3.96
Tunnel walls	4.43	18.99	13.92

Table A.2: Y+ values for mesh 3.

Part	Min. Y+	Max. Y+	Avg. Y+
Rear Wing	0.0009	12.89	3.38
Front wing	0.013	9.77	2.18
Winglets	0.008	9.44	1.9
Chassis	0.0001	11.95	3.05
Ground	0.075	18.44	4.23
Tunnel walls	5.25	18.99	13.93

Table A.3: Y+ values for mesh 4.

Part	Min. Y+	Max. Y+	Avg. Y+
Rear Wing	0.0009	12.89	3.38
Front wing	0.013	9.77	2.18
Winglets	0.008	9.44	1.9
Chassis	0.0001	11.95	3.05
Ground	0.075	18.44	4.23
Tunnel walls	5.25	18.99	13.93

Table A.4: Y+ values for mesh 5.

Part	Min. Y+	Max. Y+	Avg. Y+
Rear Wing	0.0003	13.22	2.2
Front wing	0.013	9.64	2.15
Winglets	0.005	9.53	1.95
Chassis	0.0001	11.5	3.04
Ground	0.077	18.84	4.25
Tunnel walls	4.53	18.99	13.92